Robotics and Autonomous Systems 62 (2014) 1531–1548

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Automatic generation of biped locomotion controllers using genetic programming

Pedro Silva[a], Cristina P. Santos[a,*], Vítor Matos[a], Lino Costa[b]

[a] Industrial Electronics Department, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal

[b] Production Systems Department, School of Engineering, University of Minho, 4710-057 Braga, Portugal

HIGHLIGHTS

- Automatic generation of biped locomotion with minimal knowledge about the task.
- Inclusion of sensory information enables adaptation of the locomotion.
- Results demonstrate the relevance of feedback in adapting locomotion to sloped ground.
- Inclusion of sensory information generates locomotion more entrained with the robot.

ARTICLE INFO

Article history:
Received 27 July 2013
Received in revised form
3 May 2014
Accepted 16 May 2014
Available online 5 June 2014

Keywords:
Central pattern generator
Genetic programming
Legged locomotion
Adaptation
Sensory information

ABSTRACT

Generating biped locomotion in robotic platforms is hard. It has to deal with the complexity of the tasks which requires the synchronization of several joints, while monitoring stability. Further, it is also expected to deal with the great heterogeneity of existing platforms. The generation of adaptable locomotion further increases the complexity of the task.

In this paper, Genetic Programming (GP) is used as an automatic search method for motion primitives of a biped robot, that optimizes a given criterion. It does so by exploring and exploiting the capabilities and particularities of the platform.

In order to increase the adaptability of the achieved solutions, feedback pathways were directly included into the evolutionary process through sensory inputs.

Simulations on a physic-based Darwin OP have shown that the system is able to generate a faster gait with a given stride time with improved gait temporal characteristics. Further, the system was able to cope with tilted ground within a specific range of slope angles. The system feasibility to generate locomotion more entrained with the environment was shown.

1. Introduction

There is an increasing interest in building autonomous systems to aid humans performing tasks in a wide variety of situations. Ranging from space and deep ocean exploration, or rescue missions in hazardous environments, to in everyday tasks, such as cleaning the house or taking care of the elderly. In most of these cases, legged locomotion may provide for an advantage over wheeled or tracked robots. It offers a higher level of flexibility required in a wide variety of terrains and the ability to deal with harsher terrain features, e.g. stairs, obstacles, uneven or irregular terrain. Particularly, biped locomotion provides the flexibility to a world shaped for humans. The control and generation of biped locomotion for the ever improving biped robots is a very demanding task, addressing complex problems as the generation of the movements and coordination between many degrees of freedom, balancing, perception and planning, and disturbance rejection.

Typical solutions to the problem of biped locomotion make extensive use of the knowledge of the robot and environment. Generally a plan of the path and foot placement sequence is determined, then the required motions are computed using the robot's kinematical model, respecting determined constraints established through some stability criterion, as the popular Zero-Moment Point [1,2]. However, such approach requires a good perception of the environment which may hamper the general application to different dynamic environments.

Alternatively to these typical solutions, bio-inspired approaches have been researched and proposed with quite successful results. One of these approaches uses the concept of Central Pattern

[*] Corresponding author. Tel.: +351 914730808.
E-mail addresses: psilva@dei.uminho.pt (P. Silva), cristina@dei.uminho.pt (C.P. Santos), vmatos@dei.uminho.pt (V. Matos), lac@dps.uminho.pt (L. Costa).

http://dx.doi.org/10.1016/j.robot.2014.05.008
0921-8890/© 2014 Elsevier B.V. All rights reserved.

Generators (CPGs), exploiting the interesting characteristics of intraspinal neural networks in vertebrates [3]. These generate rhythmic activation for walking motor patterns. CPGs have been widely used to address the biped locomotion generation problem [4–8] for several commercial robotic platforms (*e.g. DARWIN-OP and NAO*). The main characteristic that motivates for the application of CPGs in the generation of robotic legged locomotion is the ability to adapt and correct the locomotion by the integration of sensory feedback pathways [9,10]. This provides the ability for the robots to deal with unexpected disturbances and more dynamic, not completely known environments. Hybrid approaches include CPGs and Center of Pressure (CoP) [11] or ZMP [12].

Previously, we have proposed a CPG based solution for biped locomotion [5]. It combines a small set of motion primitives within CPGs driven by phase oscillators, producing basic but very capable biped walking for the DARwIn-OP humanoid robot. Despite the simplicity of the solution, the expansion of the repertoire of motion primitives to broaden the locomotor behaviors has proven complex, as well as the design of feedback mechanisms for the adaptation and correction of locomotion. Some authors tackle this adaptation problem through imitation and learning from demonstration [13], optimization of parameterized trajectories [14,15], for instance, by applying Genetic Algorithms (GA) [16,17], Genetic Programming (GP) [18,17] or reinforcement learning [19–21]. Besides, in [22] techniques in evolutionary robotics were used to explore the potential of a purely reactive, linear controller to control bipedal locomotion over rough terrain. This work is specially important in the sense that is able to achieve robust bipedal walking without including oscillatory neural structures.

In this work, we take a distinct approach. Our assumption is that adaptability is a very important characteristic of any controller if the system is expected to continuously adapt its behavior to the surrounding environment. This is an essential aspect if the robot is expected to cope with uncontrolled, unpredictable and dynamic worlds. In order to achieve this, a self-tuning movement is necessary as a response to the environment's properties. Therefore, the motor action should be driven by the sensory input, through feedback pathways that couple the movement generation to the environment perception. Besides, our assumption is that movement is generated through the combination of simple, brick primitives, herein considered discrete and rhythmic implemented as sinusoidal and bell-shaped motion primitives. The number, combination and tuning of these primitives need to be determined, and will result in different motor behaviors.

One of the difficulties was to specify primitives for each joint, such that the movement resulted in a gait with a given stride length. We took inspiration from the determinants of human walking [23], but it remains the question of whether the used combination of primitives was the most adequate to generate that particular gait. Therefore, herein authors apply GP to determine the primitives which optimize a given criterion. Normally, in the literature, the most used criterion are related to displacement, energy and jerk minimization. Herein, we decided that displacement is a good criterion since in order to generate a gait with a given stride length, the different joints and environment will have to be fully synchronized.

Regardless of the type of environment, locomotion is achieved by the rhythmic contraction of muscles attached to limbs, wings, fins, etc. Typically, a gait is efficient when all the involved muscles contract and extend with the same frequency and different phases [24]. Ijspeert et al. [25] concluded in their studies of the salamander locomotion that "from a dynamical systems point of view, locomotion becomes the limit cycle behavior of the controller-body-environment system".

Prokopenko [26] has shown that maximal coordination (measured information theoretically) was achieved synchronously with fastest locomotion (a direct measure). He showed that actuators are well-coordinated in individuals with fastest locomotion.

Evolutionary Computation (EC) algorithms rely on the concept of Darwin's evolution theory to find optimized solutions for a target problem, such as Genetic Algorithms (GA) and Genetic Programming (GP). The former considers a control policy whose configuration is evolved as a string of chromosomes—configuration parameters for a target problem. The latter evolves a complete control program for the task at hand. These methods use a fitness function that evaluates the candidate solutions, or individuals, and whose value is used as quality measure for a set of evolutionary operators (selection, crossover and mutation).

Candidate solutions in GP, or individuals, can fully describe the solution to the target problem, not requiring any a-priori structure. Therefore, although the complexity of the search space is increased, it is expected to generate more adequate solutions to a particular problem.

The idea of applying Evolutionary Algorithms for generating gait controllers for bipeds is not new. However, up to our knowledge, usually these works address parametric optimization, and not so much the optimization of the structure of the gait controller. In such case, it is required a more flexible scheme than the GA binary representation.

We apply GP to automatically search the solution landscape and find solutions that rely on a set of motion primitives. We also explore the inclusion of feedback pathways, through sensory inputs, as leaves of the GP tree, as a means to enable adaptation to the environment features, particularly to adapt the locomotion to walk up and down slopes in the environment. Specifically, in this work the goal is to apply GP to the automatic exploration of: (1) the motion primitives within the CPG, and (2) the integration of sensory inputs into feedback mechanisms for the adaptability to the environment.

Different simulations with a physics based simulator involving a commercially available robotic platform enable to assess the different aims. Firstly, it is considered the role of optimization and the role of the external feedback modulation in improving the gait temporal features, such as velocity, stride length and energy. We are particularly interested in the impact of sensory inclusion in the robot behavior. A functional analysis including roll, pitch, the Center of Mass (CoM) and Center of Pressure (CoP) trajectories, force sensors and ground clearance assess robot behavior. This provides for an understanding of how feedback enhanced the locomotion skills of a biped robot.

Next, the adaptation of the system to tilted surfaces is explored. For that, a case study is shown that assesses the viability of the biped system to be able to walk adaptively on tilted ground and the capability of the controller to provide more stable gaits on upslope/downslope terrains within a specific range of slope angles. Finally, a more complex simulation, in which the robot faces terrains with positive, zero and negative slopes, is implemented to verify the feasibility of the methodology to broad the locomotion.

Note we aim to verify the role of adaptability, achieved through feedback, in movement generation in irregular terrain, considering the achieved velocity. For that we have chosen a stable and slow gait, with a given stride duration, for which both approaches (including and not the feedback inputs) are expected to generate feasible locomotion controllers. The selected stride duration was based on previous work of the authors [5] in which a hand-tuned CPG architecture was able to stably walk with that stride duration. The fact that we use a long stride duration enables to generate solutions that provide for a stable locomotion. This stability facilitates the comparison of the achieved results in terms of the functional analysis. Future work could include determining which was the best stride duration for which feedback enhanced the improvement in the velocity. However, this was out of the scope of this paper.

Results demonstrate the smooth locomotion achieved by the proposed GP mechanism and the added adaptability to the environment, provided by the inclusion of feedback pathways directly onto the controller. Therefore, movement is generated in entrainment with the environment.

The paper is organized as follows. The following section presents the related work. Section 3 introduces the locomotion model used to control the target platform. Then in Section 4 the GP evolution mechanism is presented, where the individuals for the current evolution process and the evolution configuration are defined. Lastly, the results are presented in Section 5, followed by a discussion in Section 6 and conclusions and future work in Section 7.

2. Related work

GP has proven to be useful in the generation of locomotion for very different types of robotic platforms, thus showing its efficiency in finding solutions for problems with a high level of complexity.

In [27] a generic controller to an animated physically plausible 3D character, an articulated lamp, was created. In [28] a locomotion controller to an articulated, snake like robot, was created. GP was also employed to generate a legged locomotion controller for a quadruped robot in [29].

In [30], Linear Genetic Programming (LGP) was used to generate a locomotion controller with feedback pathways, for a robust and generic anthropomorphic biped robot model. The model is simulated but physically plausible, although there is no equivalent in the real world. There was no a-priori knowledge about the mechanical or physical properties of the body. The evolution used feedback from several sensor modalities (e.g. joints and several accelerometers in the body and in the legs) to successfully achieve biped locomotion. However, as described in [31], an attempt to move on to the real robot failed.

The idea of applying GP together with the concept of CPGs was also explored in [32]. GP was applied to induce parts of a "nervous system" for a 3d simulated bipedal robot, with 32 muscles that controlled rigid segments of the legs, body and arms. The neural system achieved through neural oscillators, represented a rhythm generation mechanism. The main concern of the work was the creation of the feedback networks between the neural system and the body dynamics system. Parameters and structures within single neural oscillators were assumed to be known and fixed. However, they managed to generate locomotion during only four steps. In [33], these results were improved by applying an enhanced adaptive mutation operator that reduced the search space and improved the evolution results, increasing the generated steps to ten. Although this work yielded interesting results, it uses a very specific model, which physical and mechanical properties that do not fit common biped robotic platforms.

In this work, we are interested in developing locomotion for a 3-dimensional rigid body robot, that exists commercially. This way the developed ideas can be later deployed on the real robot. Therefore, though the work herein described is implemented in simulation, it is a physics based simulator and the locomotion is implemented in a robot that is free to move in a 3D world, respecting constraints of the real world. This is a very important aspect of the developed work.

3. Biped locomotion model

The basis of the locomotion controller used in this work was previously presented in [5], where we proposed a Central Pattern Generator (CPG) integrated with local sensory feedback, capable of generating bipedal locomotor behaviors, such as walking forward/backwards and turning.

The proposed CPG controls a single leg, divided in rhythmic and unit motion pattern generators. The use of a phase oscillator as a rhythmic generator allowed for a simple contralateral coupling between the left and right CPGs, maintaining the correct coordination between the generated locomotor trajectories in both legs by producing each a driving rhythmic signal ϕ_i in strict alternation (i for left/right leg),

$$\dot{\phi}_i = \omega + k \sin\left(\phi_i - \phi_o + \pi\right), \tag{1}$$

where $\phi_i \in [-\pi, \pi]$ (rad s^{-1}) is the phase of the oscillator, increasing monotonically and linearly with rate ω.

Motion pattern generators receive this rhythmic input and produce the corresponding joint trajectory, z, in a synergistic approach of modular motion primitives encoded as a set of non-linear dynamical equations with well defined attractor dynamics, similarly to other works [34,13,35]. Basic parameterized motion primitives, f_j^m, *e.g.* sine and bell-shaped motions (implemented as Gaussians), were considered.

The joint angle value $z_{i,j}$, for leg i and joint j, is given by

$$\dot{z}_{i,j} = -\alpha\left(z_{i,j} - O_{i,j}\right) + \sum f_j^m\left(z_{i,j}, \phi_i, \dot{\phi}_i\right). \tag{2}$$

The final motor program in a single joint results from the sum of motion primitives f_j^m around a center point $O_{i,j}$. α is a relaxation parameter for the offset. j specifies the joint: hip roll (hRoll), hip yaw (hYaw), hip pitch (hPitch), knee (kPitch), ankle roll (aRoll) and ankle pitch (aPitch); i specifies the left or right leg.

3.1. Motion primitives

The proposed motions for basic bipedal walking considered in this work are herein presented in such an order, that by sequentially adding them, one can easily tune to achieve walking [23].

In this work we resort to sinusoidal and bell-shaped motion primitives. For maintaining the feet parallel to the ground at all times, we assign symmetric motions to the ankle from those performed in the hip and knee.

The motion primitives are now briefly presented.

As the robot steps alternately, it must displace the body over the supporting leg during the step cycle, allowing the contralateral leg to execute the swing phase of the step. This is achieved by *the balancing motion*

$$f_{\text{hRoll}}^{\text{balancing}} = -A_{\text{balancing}}\,\omega \sin\left(\phi_i\right) \tag{3}$$

$$f_{\text{aRoll}}^{\text{balancing}} = -f_{\text{hRoll}}^{\text{balancing}} \tag{4}$$

where i specifies the left or right leg, ϕ_i is the phase of left or right CPG, and parameter $A_{\text{balancing}}$ specifies the amplitude of the lateral displacement motion.

Leg *flexion motion* is performed by the unloaded leg so the foot achieves vertical clearance during the swing phase of the step, executed in strict alternation between the two legs. This is shown in Eq. (5) for hip, Eq. (6) for knee, and Eq. (7) for ankle

$$f_{\text{hPitch}}^{\text{flex}} = \frac{A_{\text{hip}}\,\omega\phi_i}{\sigma^2} \exp\left(-\frac{\phi_i^2}{2\sigma^2}\right) \tag{5}$$

$$f_{\text{kPitch}}^{\text{flex}} = -\frac{A_{\text{knee}}\,\omega\phi_i}{\sigma^2} \exp\left(-\frac{\phi_i^2}{2\sigma^2}\right) \tag{6}$$

$$f_{\text{aPitch}}^{\text{flex}} = -\left(f_{\text{hPitch}}^{\text{flex}} + f_{\text{kPitch}}^{\text{flex}}\right). \tag{7}$$

A_{hip} specifies the amplitude of the bell trajectory for the hip and A_{knee} for the knee. The trajectory described by the ankle is the sum of the hip and knee so the feet remains parallel at all times. This flexion motion is centered at $\phi_i = 0$, comprising the range from $\phi_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ by setting $\sigma = \frac{\pi}{6}$, performing an overall swing phase of about 50% of the step cycle.

In level human walking the pelvis rotates alternately [23], flattening the vertical COG trajectory, as well as smoothing the inflections when changing the vertical direction of the COG. This *pelvic rotation* is performed at the hip yaw joints, described as a sinusoidal trajectory with amplitude A_{rotation} and $\frac{\pi}{2}$ phase shift:

$$f^{\text{rotation}}_{\text{hYaw}} = -A_{\text{rotation}}\omega \sin\left(\phi_i + \frac{\pi}{2}\right). \tag{8}$$

The final motion we consider is responsible for producing the propulsion of the body during locomotion. This motion moves the legs in the sagittal plane, alternately moving the contralateral legs forward and backward, like a compass. This motion is described as sinusoidal profiles at the hip and ankle pitch joints, with amplitude A_{compass} and $\frac{\pi}{2}$ phase shift:

$$f^{\text{compass}}_{\text{hPitch}} = -A_{\text{compass}}\omega \sin\left(\phi_i + \frac{\pi}{2}\right) \tag{9}$$

$$f^{\text{compass}}_{\text{aPitch}} = -f^{\text{compass}}_{\text{hPitch}}. \tag{10}$$

The overall result of all the motions is straight bipedal walking. Here we summarize all the motor programs assigned to each joint:

$$\dot{z}_{\text{hRoll}} = -\alpha(z_{\text{hRoll}} - O_{\text{hRoll}}) + f^{\text{balancing}}_{\text{hRoll}}, \tag{11}$$

$$\dot{z}_{\text{aRoll}} = -\alpha(z_{\text{aRoll}} - O_{\text{aRoll}}) + f^{\text{balancing}}_{\text{aRoll}}, \tag{12}$$

$$\dot{z}_{\text{hYaw}} = -\alpha(z_{\text{hYaw}} - O_{\text{hYaw}}) + f^{\text{rotation}}_{\text{hYaw}} \tag{13}$$

$$\dot{z}_{\text{hPitch}} = -\alpha(z_{\text{hPitch}} - O_{\text{hPitch}}) + f^{\text{flex}}_{\text{hPitch}} + f^{\text{compass}}_{\text{hPitch}} \tag{14}$$

$$\dot{z}_{\text{kPitch}} = -\alpha(z_{\text{kPitch}} - O_{\text{kPitch}}) + f^{\text{flex}}_{\text{kPitch}} \tag{15}$$

$$\dot{z}_{\text{aPitch}} = -\alpha(z_{\text{aPitch}} - O_{\text{aPitch}}) + f^{\text{flex}}_{\text{aPitch}} + f^{\text{compass}}_{\text{aPitch}}. \tag{16}$$

We perform this parametrization by incrementally tuning and adding each motion in sequence by trial and error. Optimization works are currently being developed [15] to determine these parameters based on some criterion.

Further details about this approach can be found in [5].

3.2. GP application

This approach, despite simplistic, allowed a small humanoid DARwIn-OP robot to perform stable locomotion [5]. The simplicity of the approach is also its weakness. For instance, if more complex motor programs are desired we are unaware of which motion primitives f_j^m should be employed. This problem motivates us to use automatic optimization in the present work, where the motor program is given as a result from GP evolution, given by:

$$\dot{z}_{i,j} = E_{i,j}\left(z_{i,j}, \phi_i, \dot{\phi}_i\right), \tag{17}$$

where i specifies the right or left leg, and j the hip roll (hRoll), pitch (hPitch) and yaw (hYaw), knee pitch (kPitch) and ankle pitch (aPitch) and roll (aRoll). Since the motor programs are valid for both legs, and because we use the same motor programs of the hip and knee joints in the ankle joints to maintain the feet parallel to the ground at all times, we only perform the search in four motor programs for the hip roll, pitch and yaw, and for the knee pitch. This is depicted in Fig. 1.

In this work, the phase oscillator is parameterized by ω such that the resultant locomotion has a stride duration of 3 s. E functions generated through GP evolution will exploit the joint space configuration so as to generate joint trajectories that achieve the faster locomotion within the specified frequency. The optimization aims to enable faster walking under a constant stride duration period. In order to do so the generated movement will have to be much more efficient and synchronized.

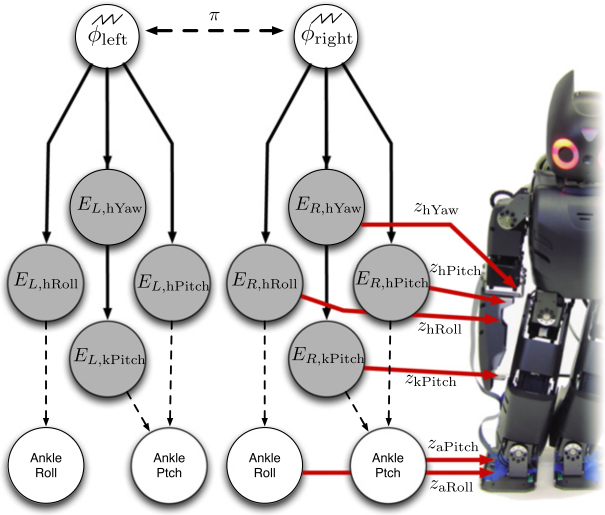


Fig. 1. Schema for the control DarwinOP using only 4 motor programs (gray dark circles).

The fact that we use a long stride duration enables to generate solutions that provide for a stable locomotion, for which both approaches (including and not sensory information) can generate useful solutions. This stability facilitates the comparison of the achieved results in terms of the functional analysis.

4. Evolution configuration

The GP evolution process aims to optimize the locomotion of a given platform, in this case the Darwin-OP biped robot. GP starts from a population of individuals that describe the locomotion of the robot. Each individual is a tree structure in which, every node has a primitive function and every terminal node has an operand. Therefore, in order to define the structure of the individuals, it is required to specify the set of terminals, the set of primitive functions and the evaluation fitness function. Iteratively, fittest individuals are probabilistically selected to generate new individuals using crossover, mutation and reproduction. Different platforms could have been used instead, under adequate configuration. Further, we are also interested in verifying how to explore movement generation in order to deal with different environments, namely to climb and/or descend a slope. In order to verify the role of the inclusion of feedback pathways in improving the gait temporal features and the functional motor behavior, sensory information is included onto the search process such that the generated movement is modulated by the acquired sensory information. This is expected to provide adaptability to environment (adaptation to climb and/or descend slopes). In summary, two specific goals are defined: (1) to improve the locomotion efficiency for the target platform comparatively to an initial hand-tuned solution by exploring different motion primitives for each joint; and (2) to explore the search and optimization of feedback pathways to achieve adaptability to changing features of the environment.

Two different controllers are proposed in order to address each of these goals: *Controller* 1 and *Controller* 2. *Controller* 1 intends to generate biped locomotion to a target platform, and evolves in open-loop without including any sensory information from the environment. *Controller* 2 intends to generate biped locomotion to the same target platform but in a closed-loop fashion. Therefore, sensory information was directly included into the movement generation in order to achieve adaptability to the environment. This last controller should therefore provide for better results when adaptability is required.

4.1. Individuals

Locomotion is generated according to four mathematical expressions E, Eq. (17). Each individual, a candidate solution for the locomotion problem, is composed by four chromosomes, that correspond to the four E equations, that will drive the robot joints, as depicted in Fig. 1.

Each chromosome is a mathematical expression under the form of a tree. The nodes of the tree are defined by functions whose branches correspond to the inputs. The leaves are variables, constants and sensory inputs, that will input the lower level tree nodes.

The function set is specified as:

$$S_F = \{\sin, \cos, *, +, -, /, \exp, \text{sigmoid}, \text{step}\}. \tag{18}$$

It is assumed that this set is sufficient to create rhythmic motions for the achievement of the biped locomotion pattern. These functions were used in [5] to implement the basic parameterized motions primitives, sine and bell-shaped motions, to achieve biped locomotion.

Further, the functions sigmoid and step were included as relevant functions for the control process. The sigmoid function was defined as follows:

$$\text{sigmoid}(x) = 2\frac{1}{1+\exp^{-5x}} - 1. \tag{19}$$

The step function according to:

$$\text{step}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{if } x < 0, \\ 0 & \text{otherwise}. \end{cases} \tag{20}$$

Both functions were thought as interesting functions to enable the interaction between functions and between functions and inputs (e.g. variables or sensory inputs).

The terminal set is defined specifically for each of the two proposed controllers.

4.1.1. Controller 1

For this controller the terminal set is defined as follows:

$$S_T = \Big\{\phi, \dot{\phi}, z, -\frac{\pi}{6}, -\frac{\pi}{4}, -\frac{\pi}{3}, -\frac{\pi}{2}, -\pi, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}, \pi, [-60, 60], 0\Big\}, \tag{21}$$

where ϕ, $\dot{\phi}$ and z are the controller inputs as defined in Section 3. The constants were chosen as angles and real values thought to be relevant for the purpose. Although the angles, such as $\frac{\pi}{2}$, are within the defined interval of real numbers, [−60, 60], their specification in the terminal set, increases the probability of being chosen. This is important for the locomotion generation due to their adequacy to define phase relations between the different joints. The real interval, [−60, 60], was chosen as possible values for the amplitudes of the different movements.

4.1.2. Controller 2

In order to be able to seek for adaptation to the environment such that locomotion is generated accordingly, sensory feedback can be directly included in the search space.

In this work, sensory feedback is provided by a three axis accelerometer, a_x, a_y and a_z; a three axis gyroscope, g_x, g_y and g_z; and the touch sensors of each leg, t_l and t_r. Both the accelerometer and the gyroscope values are normalized and then fed to the controllers. The touch sensors indicate if a given leg is in contact with the ground or not, yielding a Boolean value indicating the state.

The terminal set for evolutions is thus defined as follows:

$$S_{T_f} = S_T \cup \{a_x, a_y, a_z, g_x, g_y, g_z, t_l, t_r\}. \tag{22}$$

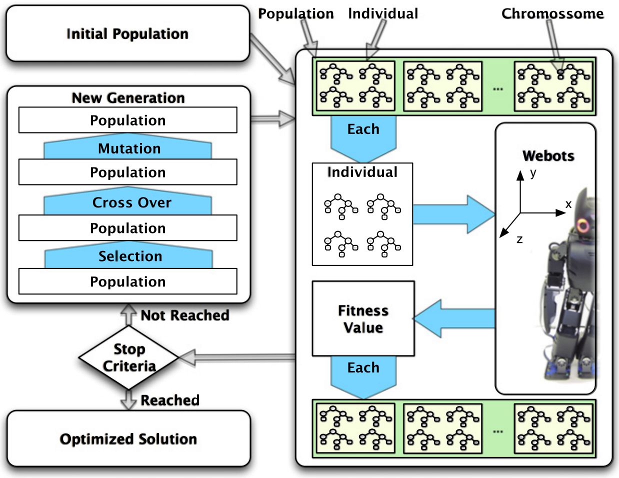


Fig. 2. Evolution architecture. On the left, a new population is generated. On the right, the evaluation process in which each individual is tested in Webots simulator.

4.2. Evaluation

The criterion used in the evaluation of the individuals is the forward distance traveled by the robot during a certain amount of time. Besides this, the fitness takes into account different and penalizable results as follows:

$$f = \frac{1}{N}\sum_{i=1}^{N}\left[\Delta_z^{(i)} - |\Delta_x^{(i)}| - c_{\text{fall}}^{(i)} v_{\text{fall}}^{(i)}\right] - c_{\text{nan}} v_{\text{nan}}, \tag{23}$$

where Δ_z is the forward displacement and Δ_x is the lateral displacement. The lateral displacement is removed from the forward displacement in order to compensate for possible asymmetric sliding of the platform. The evaluation process of each individual is divided into N stages. Each stage yields different displacements and possible falls that have to be averaged to count for the final fitness. v_{fall} and v_{nan} are flags that indicate if the controller caused the robot to fall, or produced impossible joint positions, respectively. The constants c_{fall} and c_{nan}, are the penalizing coefficients for the corresponding situations. The undesirable situations of the robot falling or attempting an impossible joint position, as well as the lateral displacement, Δ_x, penalize the corresponding individual's fitness.

4.3. Architecture

The evolution process was implemented using the OpenBeagle Framework, [36] as shown in Fig. 2. The initial population is generated with physically acceptable individuals that are evaluated using the Webots simulator. The scenario may vary and yields the fitness values accordingly. The fitness of each individual is used by the genetic operators, selection, crossover and mutation, in the process of generating a new population. The new population is then evaluated and the loop goes on until the stopping criterion is satisfied. In the present work, the stopping criterion is defined by a maximum number of generations. We used the standard crossover and mutation operators, as well as the tournament selection for the GP in OpenBeagle. The crossover and mutation operators were applied, independently, to each tree (each chromosome) of the individuals. Crossover switches one of the individual nodes with another node from another individual in the population. So, this replacement of the whole branch adds greater effectiveness to the crossover operator. The expressions resulting from crossover can be very different from their initial parents. Mutation affects an individual in the population by replacing a whole node in the selected

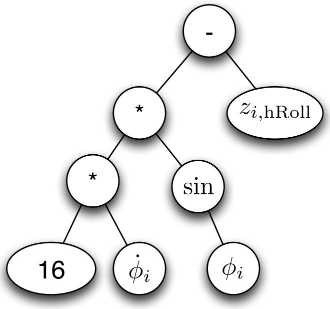


Fig. 3. Tree that implements $\dot{z}_{i,\text{hRoll}}$ primitive.

individual. In this manner, the genetic operators allow to generate solutions that fully exploit the search space.

As described in Section 3, the locomotion controller was previously developed by the team in [5], and combines a small set of motion primitives. Herein, we have built trees which replicate the required four hand-tuned motion primitives (Eq. (2)) such as to provide for a good walking gait. These trees implement the following primitives:

$$\dot{z}_{i,\text{hRoll}} = 16\,\dot{\phi}_i\,\sin(\phi_i) - z_{i,hRoll}, \tag{24}$$

$$\dot{z}_{i,\text{hYaw}} = -z_{i,hYaw} - 3\,\dot{\phi}_i\,\cos(\phi_i)\,, \tag{25}$$

$$\dot{z}_{i,\text{hPitch}} = \frac{1.9\,\dot{\phi}_i\,\phi_i}{e^{\phi_i{}^2}} - 5\,\dot{\phi}_i\,\cos(\phi_i) \tag{26}$$

$$- z_{i,hPitch} - 19, \tag{27}$$

$$\dot{z}_{i,\text{kPitch}} = 30 - \frac{109.427\,\dot{\phi}_i\,\phi_i}{e^{3.648\,\phi_i{}^2}} - z_{i,kPitch}. \tag{28}$$

Fig. 3 shows the tree build up for the $\dot{z}_{i,\text{hRoll}}$ primitive.

The initial population for all the evolutions was specified using 9 seed individuals, as variations of this hand-tuned solution, plus the hand-tuned solution itself. These seed solutions were defined by small variations in the parameters of Eq. (2), whether in the motion primitives, or in the offsets of the joints ($O_{i,j}$).

The remaining individuals were generated randomly using the ramped Half-and-Half method—where half the individuals are randomly generated trees until a variable depth is reached, and the other half until a variable size is reached. All share the same structure so that their information will be spread through the population during the evolution process, and coupled with different structures to generate novelty and diversity.

Other parameters of the evolution process configuration are listed in Table 1. The crossover, mutation and reproduction rates, as well as the selection sizes, were selected by trial and error, so that the evolution process yields optimized solutions.

The penalizing coefficient $c_{\text{nan}} = 10$, so that the genetic information that generates impossible joint positions is quickly discarded by the evolutionary process.

5. Results

This section intends to show the obtained results for the two developed controllers.

Firstly, we demonstrate the adequacy of both controllers to produce different walking locomotion with improved performance over the initial hand-tuned one, according to the specified criterion. Secondly, we explore the inclusion of sensory feedback pathways as a means to provide for adaptability and as such achieve locomotion with better performance and able to climb and descend slopes.

Table 1
GP evolution parameters.

Parameter	Value
Tournament size	10
Crossover rate	0.8
Mutation rate	0.3
Reproduction rate	0.1
Population size	500
# Generations	100
Max depth	25
Fitness function	$\frac{1}{N}\sum_{i=1}^{N}\left[\Delta_z^{(i)} - \lvert\Delta_x^{(i)}\rvert - c_{\text{fall}}^{(i)}v_{\text{fall}}^{(i)}\right] - c_{\text{nan}}v_{\text{nan}}$

Videos of the found solutions can be seen in http://asbg.dei.uminho.pt/node/346.

The obtained mathematical expressions present some degree of complexity and the parameters of the different sensory inputs can vary according to the type of function used. Thus, it is not feasible to present a specific analysis of the parameters of the evolved controllers. It is important, however, to assess which feedback loops were selected and how they achieve behavior. In order to so, other than looking to the controller parameters, an extensive analysis of the explicit final result of each controller is done. This was done through a functional analysis of the obtained gaits which aims to conclude on the impact of sensory feedback inclusion on the robot behavior considering roll, pitch, Center of Mass (CoM), Center of Pressure (CoP), force sensors and ground clearance. The role of the external feedback modulation in improving the gait temporal features, such as velocity, stride length and stance and swing durations is also explored.

However, we also show the sensor data of accelerometers and gyroscopes for the climbing slope. Since we are dealing with slopes, the role of the touch sensors in the feedback loops is not relevant. We end up by showing that there is effectively a relationship between the Z axis of the DARWIN accelerometer and the generated hip pitch joint trajectory, which variation depends whether the robot is climbing/descending or not the slope. Further, the achieved adaptation is driven by the current sensory information, and thus is different for the climbing and descending.

Results were obtained in an Intel i7-2600k 3.4 GHz Linux (8 GB of RAM) PC.

5.1. Experimental setup

The Darwin-OP is a lightweight biped robot with 20° of Freedom (DOFs). In this work, 12 DOFs, 6 in each leg, are controlled. It has eight Force-sensing Resistors on the soles of the feet, and a three axis accelerometer and gyroscope in the chest.

Simulations are run in the webots simulator [37] a simulation software based on ODE, an open source physics engine for simulating 3D rigid body dynamics.

The evolution of both controllers is performed in three different scenarios. *Flat Ground* experiment is the simplest scenario. *Upslope and Down-slope Ground* experiment is a scenario in which the robot has to climb or to descend a sloped ground. *Slope Ground* experiment is a scenario in which the same solution has to cope with up and down slopes. A more complex scenario is also shown in order to verify the feasibility of the found solutions.

In order to avoid falling during the start of locomotion, we apply a slow start to the robot locomotion. During the first 10 s of an individual evaluation, the robot movements are linearly increased from amplitude values of zero towards the determined values. This provides for a smooth and stable slow start of the robot's locomotion. The slow start period is not considered for the gait evaluation that follows.

At the end of each individual evaluation, the robot is set to its initial position and rotation, such that initial conditions are similar for the evaluation of all individuals of all populations.

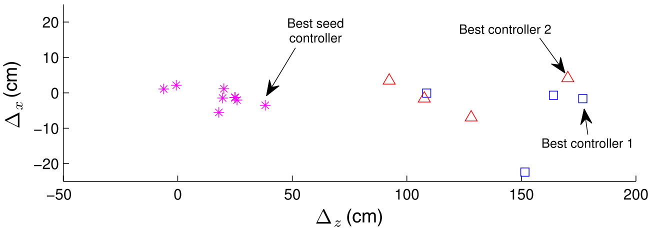


Fig. 4. Displacements achieved by 10 seed individuals (magenta stars) and 4 solutions of *Controller* 1 (blue squares) and *Controller* 2 (red triangles), in flat ground scenario. Initial position is (0, 0).

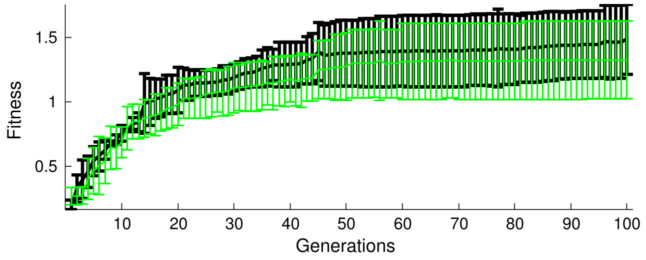


Fig. 5. Fitness Mean ± SD values for solutions of *Controller* 1 (dark black line) and *Controller* 2 (green lighter line) in flat ground scenario.

5.2. Flat ground scenario

In this scenario, the goal is to generate a controller that enables the robot to go as far as it can when moving in a flat ground during 30 s, while drifting laterally as little as possible.

The robot always moves in the same flat ground. Only one stage is considered ($N = 1$). The fall penalty factor is set as $c_{\text{fall}} = 1$, so that the evolution process is forced to select individuals that do not cause the robot to fall, over the ones that do. A constant and independent fall penalty enables the persistence of solutions that are able to walk for a while over other solutions that cannot.

Four independent evolutions were performed for each controller. Each evolution took approximately 30 h to simulate.

Fig. 4 depicts the forward and lateral displacements achieved for the 10 seed individuals (star marker) and the 4 solutions for *Controller* 1 (square marker) and *Controller* 2 (triangle marker).

The forward displacement, Δ_z, achieved by the evolved solutions of both controllers was far better than the one achieved by the seed individuals. On the other hand, the lateral displacement, Δ_x, varied within the solutions. *Best controller* 1 identifies the best solution of *Controller* 1, and has $\Delta_x \approx 2$ cm and $\Delta_z \approx 177$ cm. Comparatively to the best hand-tuned one, with $\Delta_x \approx 3.5$ cm and $\Delta_z \approx 38$ cm it improved 43% in x and 366% in z. *Best controller* 2 achieved $\Delta_x \approx 4.1$ cm and $\Delta_z \approx 170$ cm, which comparatively to *Best seed controller*, indicates an increase of 17% in x and an improvement of 347% in z.

This improvement in the forward displacement, Δ_z, is a consequence of an increase in the stride length from approximately 5 cm in the *Best seed controller*, to approximately 24 cm and 22 cm in the *Best controller* 1 and *Best controller* 2, respectively. Table 2 lists the achieved forward displacement, stride length, stance duration, velocity and consumed energy by the leg servos during locomotion.

Overall, both controllers present similar displacements. This suggests that in case there is no explicit need for adaptation, the inclusion of feedback could be discarded. However, the dispensed energy levels reveal that *Best controller* 1 required much more energy than *Best controller* 2. Further a slight difference in the stance duration is noted. *Best controller* 2 achieved a shorter stance duration than *Best controller* 1.

Fig. 5 shows the mean and SD fitness values of the solutions of *Controller* 1 (green lighter line) and *Controller* 2 (dark black line). The fitness of both controllers tends to nearly identical values at the end of the search.

The evaluation criterion of the developed controllers were based on the displacement. No considerations related to stability were taken into account in the evolution process. However, we are interested in verifying the impact of feedback inclusion on the generated locomotion. Therefore, we proceed to a functional analysis of the obtained gaits for each controller, considering pitch, roll, Center of Mass (CoM), Center of Pressure (CoP), touch sensors, force sensors, ground clearances and energy consumption.

Fig. 6 depicts the locomotion generated by *best seed controller*. Panel *A* shows the CoM (solid red) and the CoP (dashed green), alongside with the average feet positions during the stance phases of each foot (blue rectangles). Panels *B* and *C* show the Pitch and Roll angles of the robot's body, respectively. Panels *D*, *E*, *F* and *G* show the touch sensors values, the force sensors values, the measured feet height and the energy consumption, respectively, during locomotion, for the left leg. Information is similar for right foot and is not shown. A leg stance period is identified by a shaded area.

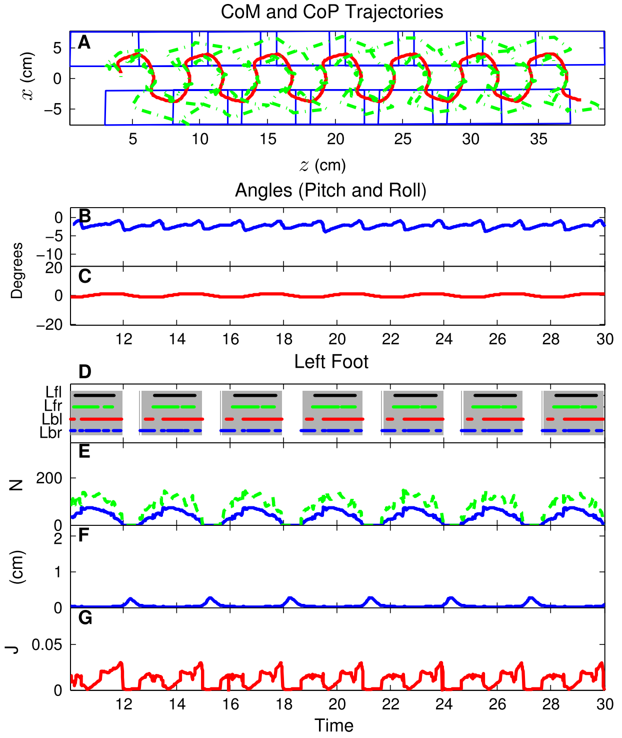


Fig. 6. Functional analysis of gait generated by *Best seed controller* in flat ground scenario. *A*: Trajectories of CoM (solid red line), CoP (dashed green) and feet positions (blue rectangles). *B*: Pitch angle of the robot's body. *C*: Roll angles of the robot's body. *D*, *E*, *F* and *G*: touch sensors values, force sensors values (mean value in blue solid line, max values in green dashed line), the measured feet height during locomotion and the measured energy, respectively, for the left leg. A leg stance period is identified by a shaded area. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We can see that the generated locomotion is quite stable. It shows smooth trajectories for the CoM and the CoP (*A*), which are coincident when the robot switches its supporting foot. The CoM is positioned inside the polygon of the robot feet during stance which provides for a good static stability. Further, the CoP is well centered within the robot foot during stance.

The stable and constant oscillation pattern observed in the Pitch and Roll angles of the robot's body during locomotion (panels *B* and *C*, respectively), agrees with this observation. This stability and locomotion smoothness is also indicated by the touch sensor (*D*) and force sensor values (*E*) that indicate stance duration with small variability. Similarly, panel *F* shows the measured vertical clearance that indicates the robot is able to lift the feet during

Table 2
Forward displacement, stride length, stance duration, velocity and energy consumption values for *Best seed controller*, *Best controller* 1 and *Best controller* 2 in the flat ground scenario.

Controllers	Δ_z (cm)	Stride length Mean ± SD (cm)	Stance Mean ± SD (s)	Velocity (cm s^{-1})	Energy (J)
Best seed	38	5 ± 0.03	2.23 ± 0.019	1.71	27.4
Best controller 1	177	24 ± 0.2	2.3 ± 0.026	8.17	88.8
Best controller 2	170	22 ± 0.23	2.03 ± 0.036	7.61	50.8

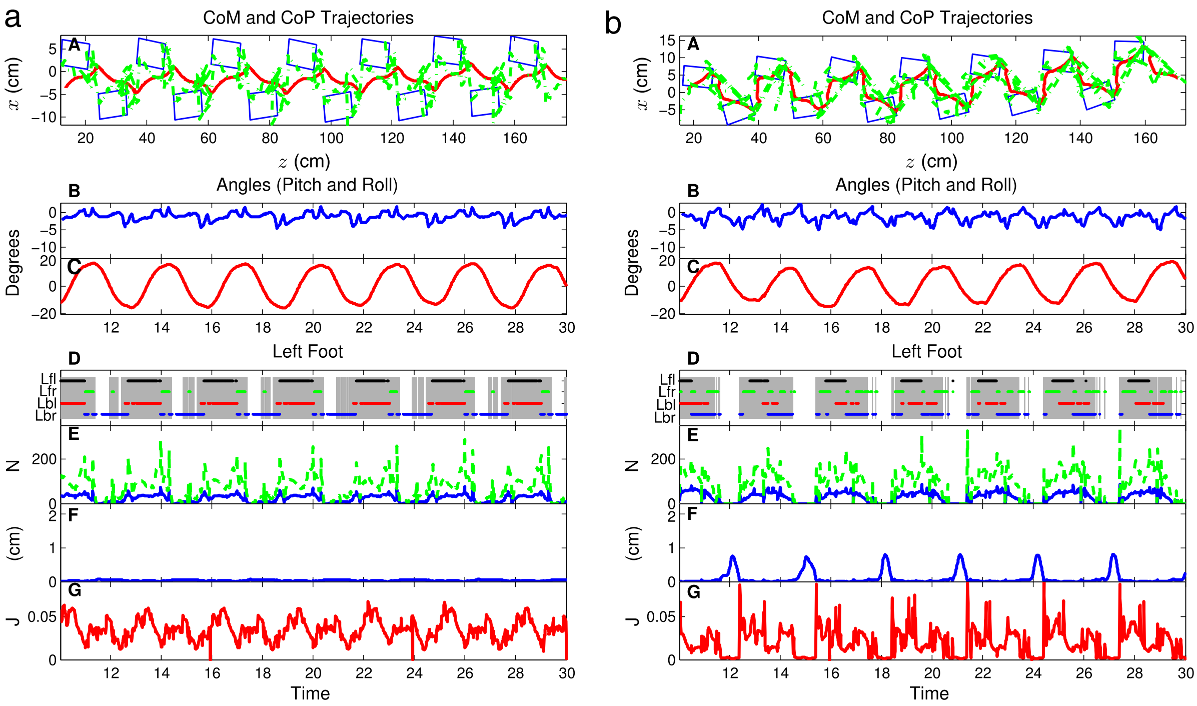


Fig. 7. Similar to Fig. 6, but for *Best controller* 1 without feedback inclusion (a) and *Best controller* 2 with feedback inclusion (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

locomotion. These values can be seen in Table 2 in which one verifies the stable pattern achieved.

A similar analysis is done in Fig. 7, for *Best controller* 1 (a) and *Best controller* 2 (b) solutions. Let us compare the different gaits achieved.

We can observe that lateral displacement is larger for the *Best controller* 2 solution and the robot turns left (Fig. 7(b), *A*). However, the inclusion of sensory information resulted in a larger excursion of the CoM trajectory. The CoP is also now within the bounds of the robot's feet thus generating a more stable locomotion. The robot weight moves towards the center of the foot which facilitates vertical clearance of the unloaded leg during the swing phase of the step. This is visible through the achieved vertical clearance (panels *F*) and the touch and force sensors' values (panels *D*, *E*). The spikes in the vertical clearance coincide with moments of no activation in the touch sensors.

On the other hand, *Best controller* 1 denotes a CoM trajectory that barely enters the support area of the robots' feet during stance (Fig. 7(a), *A*). This prevents the robot from lifting its feet during swing, thus sliding the feet during locomotion. This is shown in the touch and force sensors' values (panels *D*, *E*). At least one touch sensor is active at all times. Furthermore, in panel *F*, the measured vertical clearance also suggests that the robot slides its feet during swing. The energy dispensed in the 3 controllers is shown in panels *G*, of Figs. 6, 7(a) and (b), for *Best seed controller*, *Best Controller* 1 and *Best controller* 2, respectively. The evolved solutions dispensed much more energy during locomotion than *Best seed controller*, as was expected since they perform a much faster locomotion. *Best controller* 2 has higher spikes than *Best controller* 1, due to the highest vertical clearance, but also periods of time with lower energy consumption. In Table 2, the total energy consumption values are shown.

Overall, the inclusion of feedback seems to generate locomotion that is more entrained with the robot platform and the environment.

The aim of the evolution process was not to achieve more stable locomotion patterns, but rather to optimize the locomotion speed. The achieved results denote a huge improvement in the locomotion speed. However, the inclusion of feedback enabled to obtain solutions that besides being faster are also more stable, and more entrained with the context and platform than when no feedback was considered.

5.3. *Up-slope and Down-slope ground*

This experiment is intended to verify the adequacy of the proposed controllers to adjust the generated trajectories to the current environment, sensed by the considered sensors. This adaptation to environmental features is evaluated considering slopes, that the robot is expected to either climb or descend. So two different scenarios are herein verified, each having one stage.

The up and down slopes are depicted in 8, (a) and (b), respectively. During the evolution, the robot starts 10 cm away from the slope. Then, it advances and needs to adapt to the changing inclination level, whose maximum value is 9.8°, midway through the slope. This corresponds to a maximum height of 5.5 cm and an extension slope of ≈50 cm. Afterwards, the ground is again flat until the end of the simulation.

No hand-tuned solution was able to climb the slope, either the robot fell at the beginning of the slope, or it did not advance,

Table 3
Forward displacement, stride length, stance duration, velocity and energy consumption values for *Best seed controller*, *Best controller* 1 and *Best controller* 2 in the Up-slope and Down-slope Ground.

Controllers	Δ_z (cm)		Stride length Mean ± SD (cm)		Stance Mean ± SD (s)		Velocity (cm s^{-1})		Energy (J)	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
Best seed	22.8	57	–±	5.3 ± 1.3	–±	2.11 ± 0.24	–	1.8	–	38.9
Best controller 1	130	165	11.8 ± 0.7	14.3 ± 1.8	1.54 ± 0.184	1.73 ± 0.312	3.8	4.9	75.2	45.4
Best controller 2	133	159	12.4 ± 1.5	14 ± 1.5	2.04 ± 0.157	1.85 ± 0.127	4	4.9	71.9	45.8

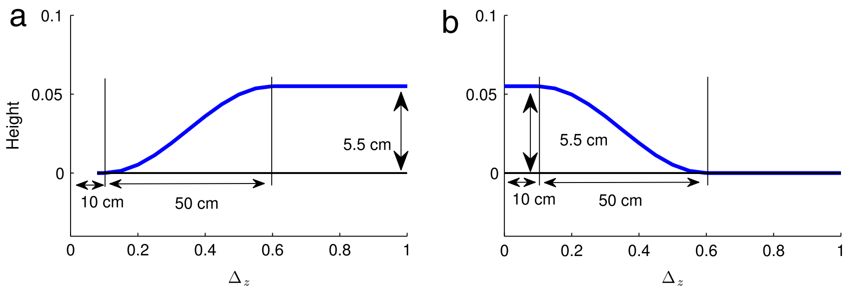


Fig. 8. Up-slope (a) and Down-slope (b) ground stages.

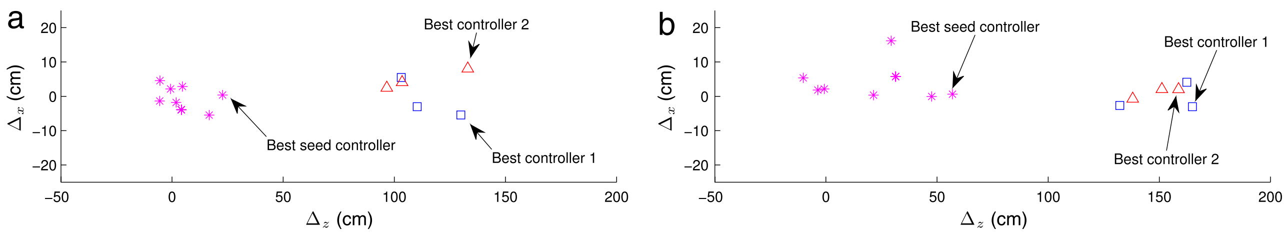


Fig. 9. Forward and lateral displacements when the robot climbs (a) and descends (b) the slopes, for 10 seed individuals (magenta stars), *Controller* 1 (square markers) and *Controller* 2 (triangle markers).

being stuck at the bottom of the slope. However, in the down-slope scenario the hand-tuned solutions were able to descend the slope. The fall penalty factor was set to $c_{\text{fall}} = 0.2$ such that the controllers that caused the robot to fall were still penalized, but those that were able to reach the slope and only fell while in the slope, could persist in the populations and still generate solutions that adapt the locomotion. As in preliminary generations solutions seemed to require more time to successfully climb or descend the slope, the evaluation time was increased to 40 s, instead of the previous 30 s.

In these scenarios, both controllers were evolved three times. Each of which took approximately 35 h to simulate.

The results are presented in Fig. 9(a) and (b), for the up and down slopes, respectively. The achieved z displacements are similar for *Best controller* 1 (square markers) and *Best controller* 2 (triangle markers) in the up slope. *Best controller* 1 has $x \approx -5$ cm and $z \approx 130$ cm. Comparatively to the *Best seed controller* with $x \approx 0.4$ cm and $z \approx 22.8$ cm it increased 1150% in x and improved 470% in z. *Best controller* 2 has $x \approx 8$ cm and $z \approx 133$ cm. Comparatively to the *Best seed controller* it increased 1900% in x and improved 483% in z.

However, in the down slope, *Best controller* 1 achieved slight higher x and z displacements ($x \approx -3$ cm and $z \approx 165$ cm) than *Best controller* 2 ($x \approx 2$ cm and $z \approx 159$ cm), similarly to the flat ground scenario. Comparatively to the *Best seed controller* with $x \approx 0.6$ cm and $z \approx 57$ cm, *Best controller* 1 (*Best controller* 2) increased 400% (233%) in x and improved 189% (178%) in z.

The climbing seems to be harder than descending since z is much smaller, as expected.

Table 3 lists the achieved forward displacement, stride length, stance duration, velocity and energy consumption for each of these controllers and scenarios.

No seed solution was able to climb. Considering the evolved solutions, for the climbing scenario, the verified improvement in the forward displacement is a consequence of an increase in the mean value of the stride length. The stride length was of ≈11.8 cm for *Best controller* 1 and ≈12.4 cm for *Best controller* 2. As expected, the dispensed amount of energy during climbing is much higher than during the descending, for the evolved solutions.

The stride length improved while descending from ≈5.3 cm in *Best seed controller* to ≈14.3 cm for *Best controller* 1 and ≈14 cm for *Best controller* 2.

Considering energy, since no seed controller was able to climb the slope, the displayed energy value for the climbing scenario is smaller than the one of the descending one. However, during climbing the dispensed energy is slightly higher when no feedback is included in the evolution, similarly to the flat ground scenario. During descending, the energy consumption results are very similar, even comparatively to the *Best seed controller*. In this scenario however, the stance periods of *Best controller* 1 are slightly longer than of *Best controller* 2.

Fig. 10 presents the mean and SD values of the fitness evolution for *Controller* 1 (dark black line) and *Controller* 2 (lightest green line) during the climbing (Fig. 10(a)) and descending (Fig. 10(b)) scenarios. This figure also shows that climbing is harder to achieve than descending, since for the second one much larger values of fitness are achieved.

A functional analysis of the achieved gait by the best solutions of each controller follows. The aim is to understand the advantages provided by the use of feedback in achieving adaptability to the different scenarios, by verifying the impact of the used sensory information in the gait features.

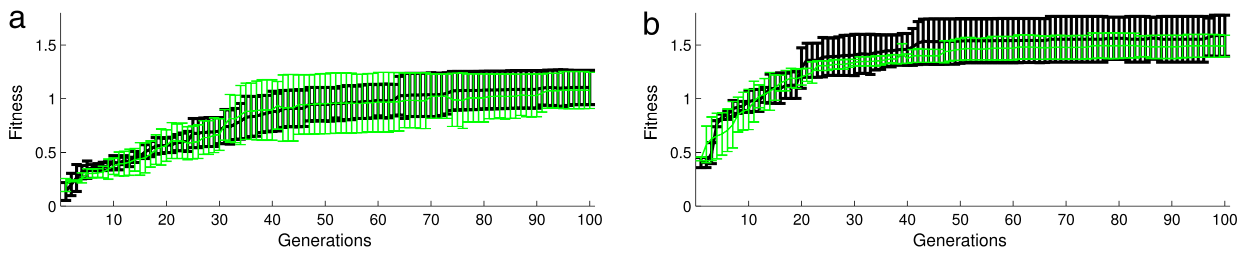


Fig. 10. Mean and SD fitness values of *Controller* 1 (dark black line) and *Controller* 2 (lightest green line) during climbing (a) and descending (b) of the slope.

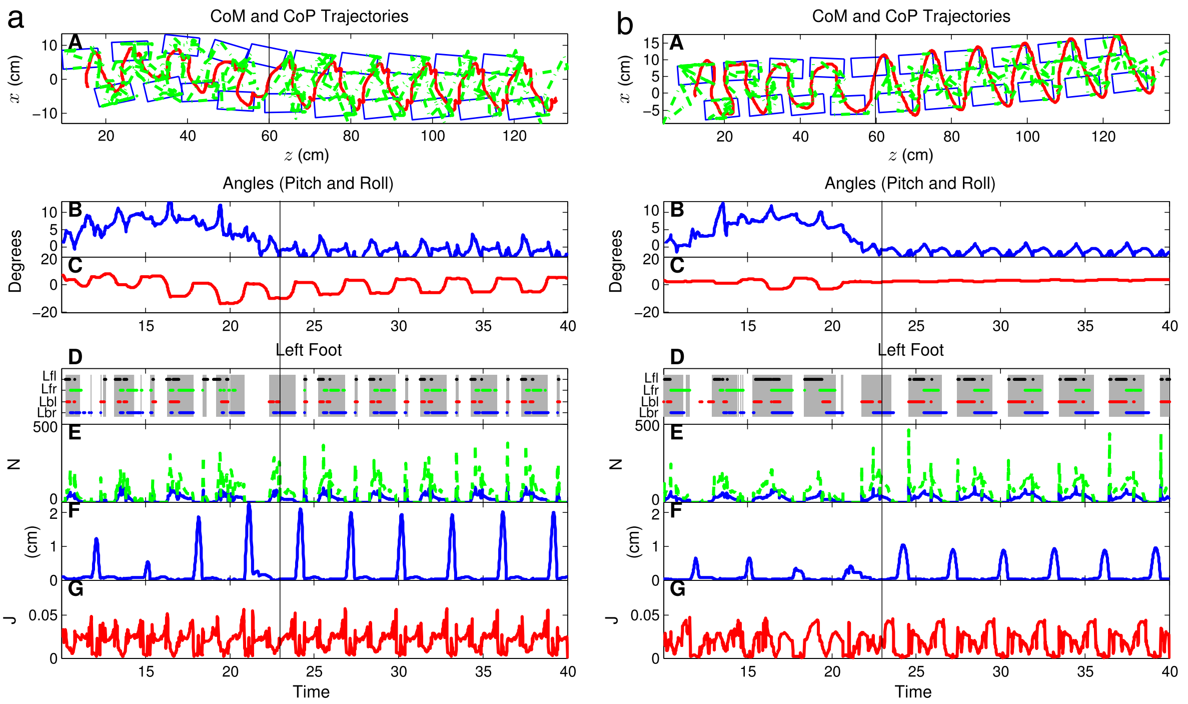


Fig. 11. Similar to Fig. 6, but for *Best controller* 1 without feedback inclusion (a) and *Best controller* 2 with feedback inclusion (b), while climbing a slope. Vertical Line indicates the moment the robot reaches the top of the slope. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 11 shows the obtained results for *Best controller* 1 (Fig. 11(a)) and *Best controller* 2 (Fig. 11(b)) solutions, respectively, when climbing a slope. The robot walks over the flat ground during $t \approx 10$ s ($t \approx 9$ s) and then it starts climbing the slope until $t \approx 23$ s ($t \approx 23$ s) for *Best controller* 1 (*Best controller* 2) solution. The robot's body pitch angle varies similarly for both controllers, according to the slope's inclination (panel B).

The roll angle (panel C) denotes a period of greater instability during the climb, which is followed by a period of larger stability. During climbing, *Best controller* 1 (*Best controller* 2) solution shows a higher oscillation amplitude with a maximum of ≈0.24 (≈0.13). After climbing, this amplitude reduces to ≈0.16 (≈0.02). Both solutions are able to rise its feet, but much less when feedback is considered (panels D to F). Further, a much more clean stance and swing phases are achieved with less glitches due to the feedback inclusion.

Best controller 1 shows a CoM trajectory with a considerable alteration of the oscillation pattern (A). Note that the center of oscillation seems to vary. It starts centered, then during climbing shifts towards the left foot, and then towards the right leg. Afterwards, when in flat ground again, it centers again. This indicates a possible reduction of the locomotion stability caused by the ground's slope.

On the other hand, *Best controller* 2, shows a period of reduced amplitude in the CoM and CoP trajectories during the climbing (A). This suggests an adaptation of the locomotion to the ground slope. Note that comparatively to when in flat ground, during climbing the CoP is more positioned backwards, as expected.

We are interested in evaluating what were the feedback loops that were selected and how do they achieve the behavior. Fig. 12 shows all the sensor data of accelerometers and gyroscopes for the climbing slope. Since we are dealing with slopes, the role of the touch sensors in the feedback loops was not relevant. We can easily verify that there is effectively a relationship between the Z axis of the DARWIN accelerometer and the generated hip pitch joint trajectory, and that the accelerometer feedback loop is the most relevant.

Fig. 13 shows the time evolution of the Z axis of the DARWIN accelerometer (top) and the generated trajectory for the hip pitch joint (middle and bottom panels). The accelerometer data changes similarly to the pitch, and one can verify the robot climbing the slope (from $t \approx 10$ to 23 s). The hip pitch joint trajectory slightly changes in accordance to the accelerometer data, and the variation depends whether the robot is climbing or not. A close up plot of the generated joint trajectory (bottom panel) shows that the change is different during climbing and afterwards it stabilizes around the −18°.

Fig. 14 shows the obtained results for *Best controller* 1 (Fig. 14(a)) and *Best controller* 2 (Fig. 14(b)) solutions, respectively,

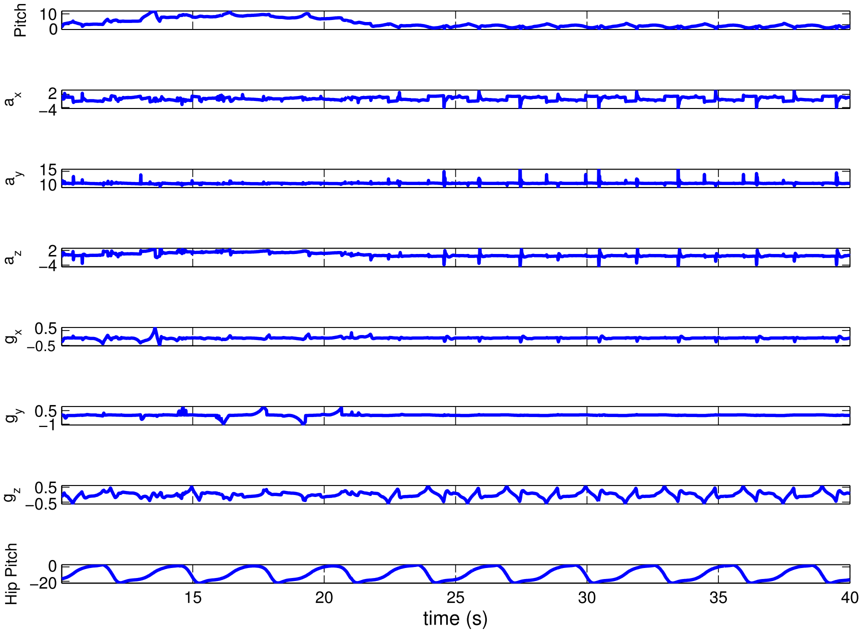


Fig. 12. Time evolution of pitch (up), robot accelerometers and gyroscopes and the left hip pitch joint (bottom) during the climbing of a slope, with *Best controller* 2 solution.

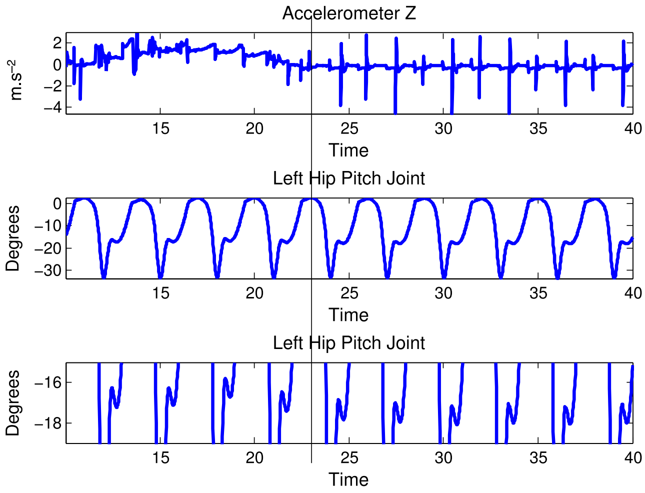


Fig. 13. Time evolution of the *Z* axis of the robot accelerometer (top) and the left hip pitch joint (middle) during the climbing of a slope, with *Best controller* 2 solution. Bottom panel shows a zoom of the hip pitch joint trajectory. Vertical Line indicates the moment the robot reaches the top of the slope.

when descending a slope. The robot walks over the flat ground during $t \approx 9$ s ($t \approx 9$ s) and then it starts descending the slope until $t \approx 22$ s ($t \approx 21$ s) for *Best controller* 1 (*Best controller* 2) solution.

The robot's body pitch angle varies similarly for both controllers, according to the slope's inclination (panel B). Both solutions are able to rise its feet during the swing phase, (panel D).

Observe the differences in the roll angle oscillations (panel C). Although *Best controller* 2 solution has a higher amplitude of oscillation (up to ≈22.9°), it denotes a constant oscillation pattern, even while descending the slope, with a slight decrease in the amplitude.

Best controller 1 solution shows a CoM trajectory (A) in which center shifts towards the left foot, when descending the slope, and then centers again in both feet in flat ground again. On the contrary, *Best controller* 2 solution presents a CoM (and CoP) trajectory that, despite a slight alteration of the oscillation pattern, keeps centered within both feet while descending the slope (A). Thus, both the CoM and CoP are kept in a safer area (centered in each foot or between both feet), contributing to the gait stability while descending the slope.

Again, this alteration of the locomotion during the most critical moment of the task, suggests the adaptability of the locomotion to the ground's slope. This is supported by an alteration of the generated locomotion for the HipPitch joints (bottom panel), according to the *Z* accelerometer value (top panel), as shown in Fig. 15. Note the slight alteration of the offset of the generated hip pitch joint trajectory (bottom panel), caused by the accelerometer data. Further, the adaptation is different from the one achieved when climbing. This shows that the adaptation is driven by the current sensory information.

A solution able to climb (descend) a specific slope could climb (descend) smaller slopes. This denotes the solution robustness to smaller perturbations.

All solutions were tested in the converse scenario, *i.e.*, if they were evolved to climb the slope, they were tested to descend it, and vice-versa. In all those cases, the robot fell. This suggests that the achieved adaptation to the slope did enable the robot to fully adapt only to the specific environment conditions. This shows up a static adaptation for a specific task.

A relevant question emerges. What if the sensory feedback pathways of the *Controller* 2 solutions were set to zero when performing the different scenarios? This would verify the need of the controller for feedback inclusion, when driving the robot through the defined task. The results have shown that solutions failed to fulfill the task. Thus, the feedback was mandatory for those solutions to achieve the necessary adaptation of the robot's locomotion.

5.4. Adaptation to slopes

In order to achieve a higher flexibility in the found solutions, namely the ability of being able of both climbing and descending a slope, the following simulations were tried. In this scenario two stages, $N = 2$, are considered. A solution is firstly evaluated in the climbing scenario and then continues to evolve in the descending one. At each stage, the robot is placed in the adequate position and orientation from similar initial conditions. The robot is forced to deal with both stages during evolution, even if it fails during one of them. By forcing the robot to be able to face both these stages, the controller needs to adapt the locomotion to the slope, rather than

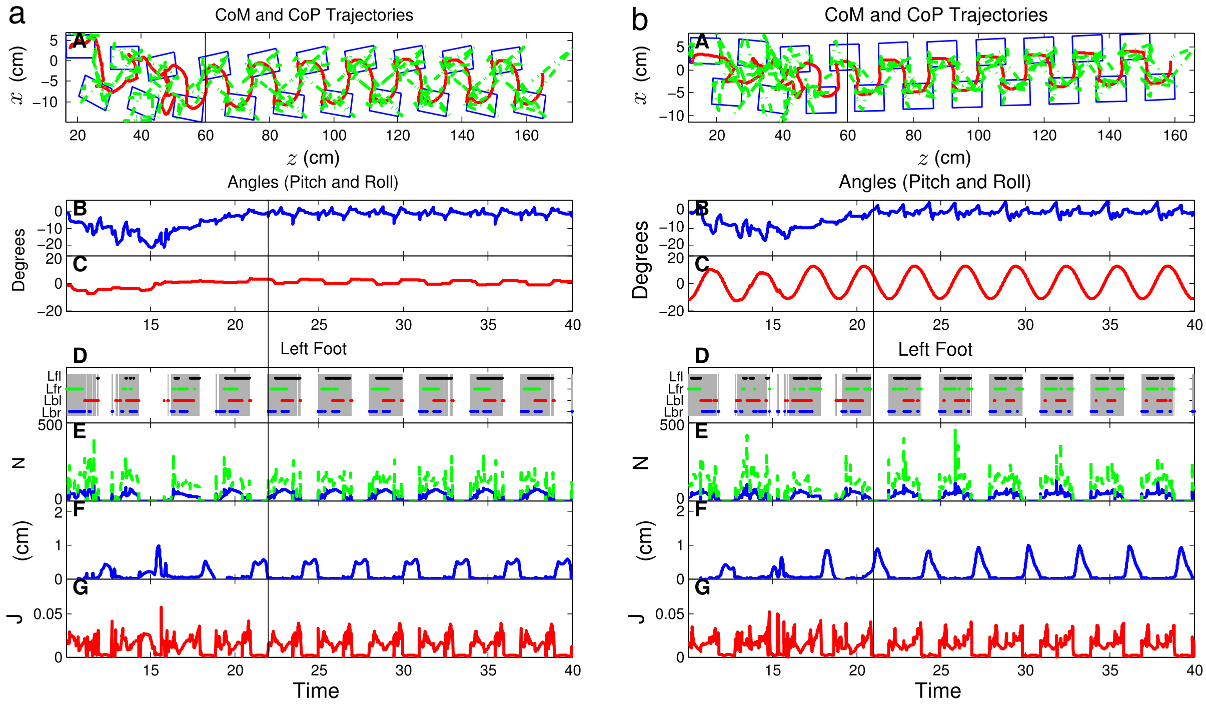


Fig. 14. Similar to Fig. 6, but for *Best controller* 1 without feedback inclusion (a) and *Best controller* 2 with feedback inclusion (b), while descending a slope. Vertical Line indicates the moment the robot reaches the bottom of the slope. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

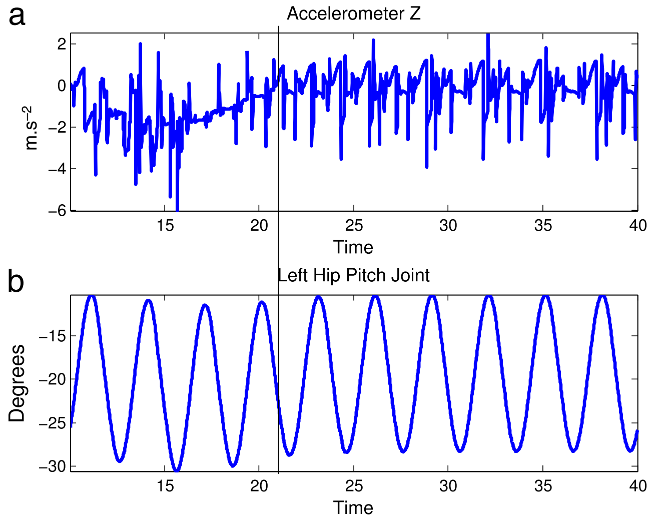


Fig. 15. Time evolution of the *Z* axis of the robot accelerometer (top) and the left hip pitch joint (middle) during the descending of a slope, with *Best controller* 2 solution. Vertical Line indicates the moment the robot reaches the bottom of the slope.

statically adapt the robot posture during locomotion. The goal is to verify if sensory inclusion provides for the required ability to adapt the generated locomotion to the environment, as well as to compare the results of both approaches. Further, we are interested in verifying the impact of sensory inclusion on the robot behavior.

In this scenario, each stage is evaluated for 40 s and/or a maximum forward displacement of $\Delta_{z_{\max}} = 80$ cm. This way, the over specialization on one of the stages is prevented. The fall penalty factor is set to $c_{\text{fall}} = 0.2$.

Three evolutions were performed for both the proposed controllers. Each took approximately 65 h to simulate.

The obtained results are listed in Table 4.

The seed solutions were not able to climb the slope but they did not fall. *Best seed controller* on this scenario achieved $\Delta_x \approx 0.4$ cm and $\Delta_z \approx 23$ cm in the climbing stage and $\Delta_x \approx -0.8$ cm and $\Delta_z \approx 21$ cm in the descending stage.

Controller 1, without the inclusion of sensory inputs, could not determine solutions able to both climb and descend the slope. The robot fell during the climbing stage. *Best controller* 1 fell in the climbing stage and achieved $\Delta_x \approx 5$ cm and $\Delta_z \approx 79$ cm when descending. This evolved solution achieved a displacement increase of 1150% in *x* and an improvement of 276% in *z* in the descending stage.

On the other hand, *Controller* 2 solutions were all able to both climb and descend the slopes. These results show that the inclusion of feedback enabled the robot to perform the task in both stages, thus adapting the locomotion as required. *Best controller* 2 was able to perform both tasks, achieving a $\Delta_x \approx -4$ cm and a $\Delta_z \approx 75$ cm while climbing and a $\Delta_x \approx -6$ cm and a $\Delta_z \approx 79$ cm while descending. Comparatively to the *Best seed controller* solution, one notes a displacement increase of 900% in *x* and an improvement of 226% in *z* in the climbing stage, and an increase of 650% in *x* and an improvement of 276% in *z* in the descending stage.

Table 5 lists the achieved forward displacement, stride length, stance duration, velocity and energy consumptions for each of these best solutions. During the descending stage, *Best controller* 1 denotes a slightly longer stance duration and energy consumption. The remaining values are similar for *Best controller* 1 and *Best controller* 2.

Fig. 16 depicts the progress of the fitness value for *Controller* 1 (dark black line) and *Controller* 2 (lightest green line), throughout the evolution process. As the forward displacement is limited to $\Delta_z = 0.8$ m and the fitness is calculated by the average of the stages, the fitness value is limited. Thus, *Controller* 2 is close to the maximum fitness value, whereas *Controller* 1 barely reaches 0.5, as a consequence of not being able to perform both stages.

A functional analysis of the *Best controller* 1 and *Best controller* 2 solutions (highlighted in Table 4) follows. These are the solutions with the highest fitness for both controllers.

Table 4
Relevant information of the three evolutions for two stages scenario for *Controller* 1, *Controller* 2 and the three best seed controllers.

	Up	Down	f	Disabled feedback	Full slope
Seed controllers	Δ_z: **23** Δ_x: **0.4**	Δ_z: **21** Δ_x: **−0.8**	**0.21**	–	*No*
	Δ_z: fall Δ_x: fall	Δ_z: 54 Δ_x: 2.3	0.15	–	No
	Δ_z: fall Δ_x: fall	Δ_z: 45 Δ_x: 2.3	0.14	–	No
Controller 1	Δ_z: *fall* Δ_x: *fall*	Δ_z: **79** Δ_x: **5**	**0.29**	–	*No*
	Δ_z: fall Δ_x: fall	Δ_z: 79 Δ_x: 12	0.25	–	No
	Δ_z: fall Δ_x: fall	Δ_z: 78 Δ_x: 6	0.27	–	No
Controller 2	Δz: 60 Δx: 5	Δz: 65 Δx: 7	0.56	No	Yes
	Δ_z: **75** Δ_x: **−4**	Δ_z: **79** Δ_x: **6**	**0.72**	*No*	*Yes*
	Δ_z: 63 Δ_x: 9	Δ_z: 65 Δ_x: 7	0.55	No	No

Table 5
Forward displacement, stride length, stance duration, velocity and energy consumption for *Best seed controller*, *Best controller* 1 and *Best controller* 2 in the staged slope scenario.

Controllers	Δ_z (cm)		Stride length Mean ± SD (cm)		Stance Mean ± SD (s)		Velocity (cm s^{-1})		Energy (J)	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
Best seed	23	21.4	–	1.87 ± 0.97	–	1.87 ± 0.408	–	0.65	–	31.4
Best controller 1	*fall*	79	*fall*	8 ± 0.8	*fall*	2.16 ± 0.286	*fall*	2.7	*fall*	35.7
Best controller 2	75	79	7 ± 1.5	7.9 ± 1.9	1.92 ± 0.259	2.03 ± 0.17	2.3	2.7	31.6	31.2

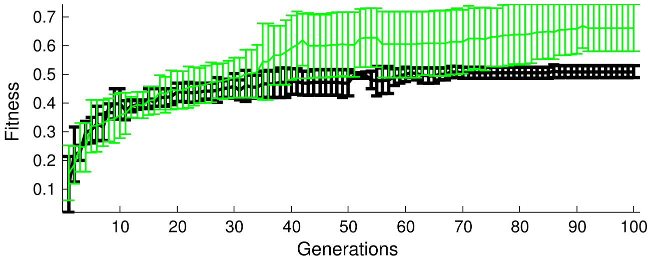


Fig. 16. Evolution of fitness value of the best *Controller* 1 (dark black line) and *Controller* 2 (lightest green line) solutions, in the slope staged scenario.

Besides finding solutions able to perform both the climbing and the descending stages without falling, *Best controller* 2 solution presents an enhanced locomotion gait over the one of the *Best controller* 1 solution. This also suggests that *Best controller* 2 is a more general solution.

Best controller 1 solution is not able to climb the stage and falls. While descending, from $t \approx 12$ s to $t \approx 31$ s, the robot maintains a stable pitch oscillation (Fig. 17, B). On the contrary, the roll angle (panel C) shows an oscillation pattern while descending the slope and then reduces this oscillation. Further the obtained CoM and CoP trajectories (Fig. 17, A) barely go in the support area of each foot, but rather are almost constantly centered in between both feet. Note that the swing phase varies considerably throughout the experiment, and the unloaded leg has a very low value of vertical clearance (F).

In *Best controller* 2 solution, the CoM and CoP movement oscillates more between the center of both feet, both during the climbing stage (Fig. 18(a), A), from $t \approx 12$ s until the point of maximum inclination, around $t \approx 24$ s; and the descending stage (Fig. 18(b), A), from $t \approx 14$ s to $t \approx 31$ s. The robot reaches the top of the up-slope at $t \approx 33$ s. The shifting of the CoM between the feet during locomotion enables the vertical clearance of the unloaded foot. Note, however, that while climbing the COM is more over the right side, and therefore, the right foot almost does not rise (panel J of Fig. 18(a)). While descending (Fig. 18(b)) the feet also denote a reduction in the spikes of ground clearance (panels F and J).

On the other hand, the force measured on the feet (panels E and I) and the spikes of feet height (panels F and J) are synchronized with moments of no activation of the touch sensors (panels D and H), while climbing and descending. This shows up the ability to achieve vertical clearance of the feet during swing.

The energy consumptions on Table 5 indicate that *Best controller* 2 solution have slightly lower results when compared to *Best controller* 1 and to *Best seed controller*. Otherwise, both controllers achieved similar locomotion results for the descending task.

This functional analysis shows up that the inclusion of feedback enabled the generation of locomotion adapted to the environment.

Such adaptation can be seen in the generated trajectories for the knee pitch joints. These trajectories show different changes according to the accelerometer data value during the climbing and descending of the slopes.

Fig. 19 shows the time evolution of the Z axis of the DARWIN accelerometer (top) and the generated trajectory for the left knee joint (bottom), during the climbing (left) and descending (right) stages. The oscillation of the accelerometer varies in accordance to the slope's inclination level during each stage. For instance, initially the sensor value starts leveled and oscillates close to 0. Then,

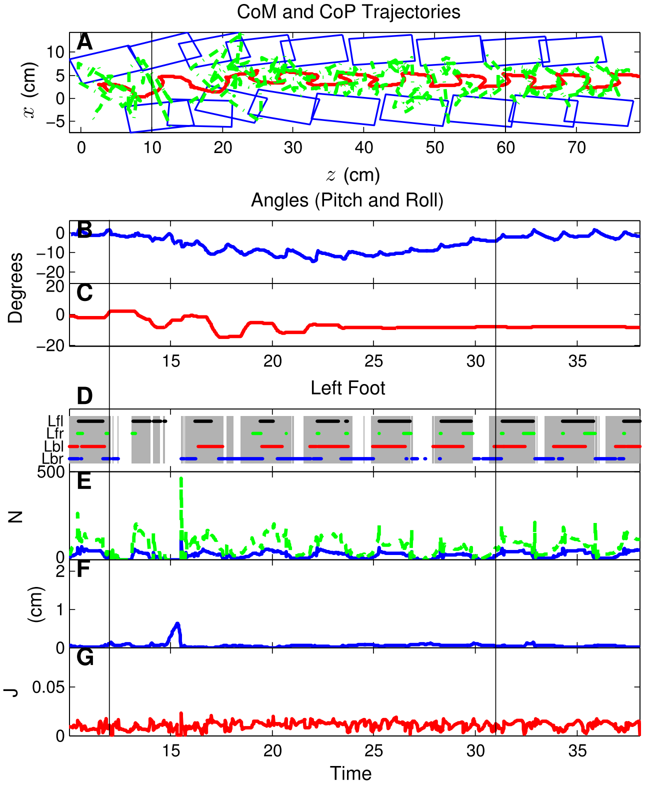


Fig. 17. Similar to Fig. 6, but for *Best controller* 1 solution without feedback inclusion, while descending a slope. Vertical Lines indicate the moment the robot starts to descend the slope and reaches its end. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as the robot starts climbing the slope (left), at $t \approx 12$ s, the offset of the oscillation increases up until half of the slope, where the inclination level is maximum. Afterwards, at $t \approx 22$ s, the offset decreases again until becoming leveled again, which corresponds to the instant of time at which the robot reaches the top of the slope, at $t \approx 33$ s.

This adaptation was also seen in previous experiment but only for a single task, *i.e.*, the robot could only adapt to the climbing or to the descending of the slope. In those cases, the feedback provided for a stabler locomotion able to achieve a single task. In this experiment, besides improving stability, the same solution is able to adapt itself to different tasks, even opposite. It adapts to a positive inclination level while it climbs and to a negative inclination level while it descends.

The inclusion of feedback on the search process, enabled the system to deal with both up and down slopes, which were not possible without the feedback. This shows up the adaptability of the proposed solution. Further, all found solutions were also able to perform locomotion in slopes with lower inclination values.

Furthermore, the solutions found when performing the up–down slope with *Controller* 2, were tested by setting the sensory inputs of the evolved controller to zero. The obtained results are presented in Table 4 in column labeled *Disabled feedback*. An *yes* indicates that the solution was still able to walk over the scenario. No solution was able to perform the stages. This shows up that feedback pathways are necessary to fulfill the task.

5.5. General experiment

In order to verify the solutions' generality a different scenario was used. During 80 s, the robot is faced with a complete sine curve, including up and down slopes, in a continuous fashion, Fig. 20. This scenario is intended to verify if the previous found solutions are able to cope with the overall path. It requires the controller to continuously adapt the locomotion as it progresses through the slope.

The achieved results are listed in the last column of Table 4, labeled *Full Slope*. An *yes* indicates if the solution was able to walk over the scenario. One solution of the *Controller* 2 solutions was not able to walk over the overall path. None of the *Controller* 1 or seed solutions was able to perform the full slope scenario. One of the seed solutions did not fall but stayed blocked in the same position.

Fig. 21 shows the achieved trajectories for the *Best controller* 2 solution (highlighted in Table 4). At $t \approx 13$ s the robot starts climbing the slope up to $t \approx 35$ s, at which time it reaches the top of the slope. Then it starts descending until $t \approx 57$ s, at which time it starts walking over flat ground again.

Despite the final lateral displacement, it successfully achieves a forward displacement $\Delta_z \approx 170$ cm. The pitch angle (panel B) shows the inclination level the robot is being faced with. The roll angle (panel C) depicts an almost fully stable oscillation pattern, only slightly perturbed near the top of the slope, at $t \approx 35$ s.

Fig. 22 presents the time evolution of the *Z* axis of the DARWIN accelerometer (top) and the generated trajectory for the left knee joint (bottom). The Knee joint trajectory changes according to the accelerometer data. In this scenario, the achieved adaptation is continuous throughout the slope.

Fig. 23 presents the total energy consumption levels, and the Mean ± SD of the stance duration and ground clearance values, for the different controllers and scenarios presented in this article. In the up stage of the last scenario (the one where the robot needs both to climb and descend the slope), the results of *Controller* 1 are not displayed since the robot was not able to fulfill this task.

We can see that in general the inclusion of feedback enables the generation of more efficient locomotion controllers with lower energy levels. This difference is more pronounced in flat ground and up-slope scenarios.

Considering the flat ground scenario, the inclusion of feedback enabled to achieve lower mean values of energy consumptions, with much lower variability. The achieved ground clearance values for *Controller* 2 are also higher, thus revealing a better ability to lift the feet during locomotion. No similar differences were found for both controllers.

In the up-slope, the inclusion of feedback also enabled to achieve lower mean values of energy consumptions, but with higher variability. Nonetheless, the maximum of *Controller* 2 is lower than the minimum of *Controller* 1, which means that *Controller* 1 solutions always spend more energy than those of *Controller* 2. Ground clearance values are higher for *Controller* 1 and with higher variability.

Considering the down-slope scenario, both controllers achieved similar energy consumption levels, as well as similar ground clearance and stance phase durations.

In the staged scenario, it is relevant to stress, that *Controller* 1 could not perform the climbing stage. Thus, it evolved towards becoming an expert in the descending stage. On the other hand, *Controller* 2 evolved to cope with both stages, thus generating a more general locomotion solution. During the descending stage of the staged scenario, *Controller* 1 shows slightly lower energy consumptions.

Overall, ground clearance results reveal that most solutions achieve ground clearance, though with smaller values in the flat ground scenario.

Further, results indicate that, as expected, the climbing stage requires more energy than the descending one. Stance duration values are rather similar for both stages. Only in the descending stage, *Controller* 2 shows slightly shorter stance duration periods than *Controller* 1.

Finally, Table 6 surveys the feedback loops that were selected for the different sloped scenarios. As described in the text, the most relevant was the *Z* axis of the robot accelerometer.

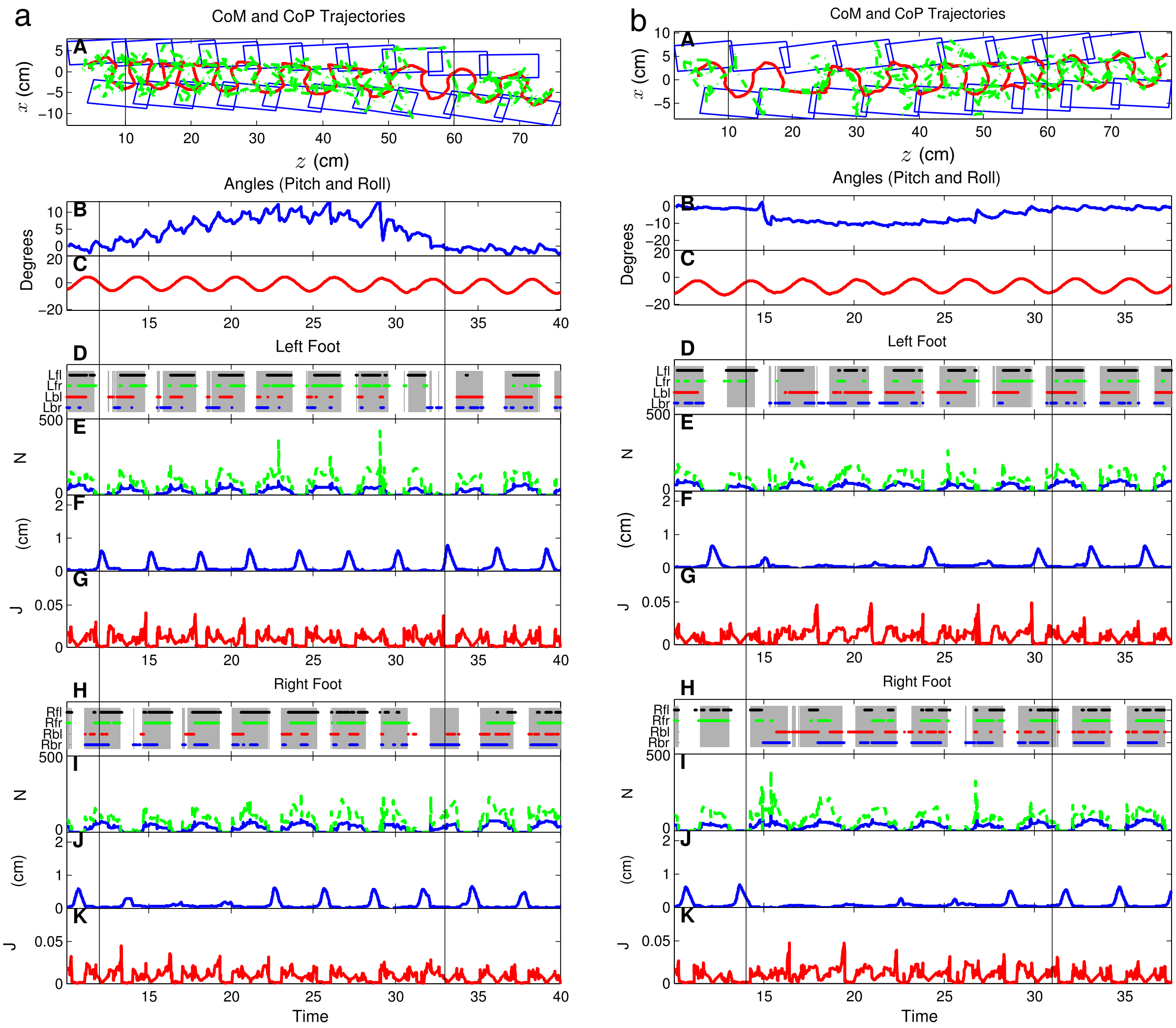


Fig. 18. Similar to Fig. 6, but for *Best controller* 2 solution with feedback inclusion, while climbing (a) and descending (b) a slope. Vertical Lines indicate the moments the robot starts to climb (descend), and reaches the top (bottom) of the slope, respectively from left to right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

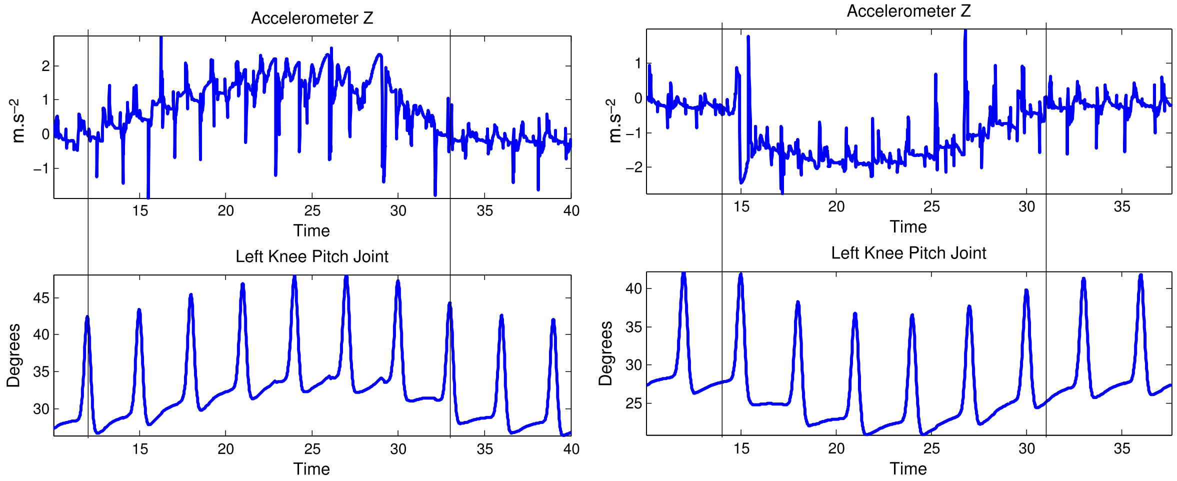


Fig. 19. Time evolution of the *Z* axis of the robot accelerometer (top) and the left knee pitch joint (bottom) during the climbing (left) and descending (right) of a slope, with *Best controller* 2 solution. Vertical Lines on the left (right) indicate the moments the robot starts to climb (descend) and reaches the top (bottom) of the slope, respectively from left to right.

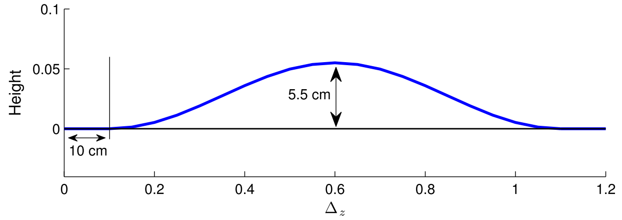


Fig. 20. Complete curve for the robot to traverse.

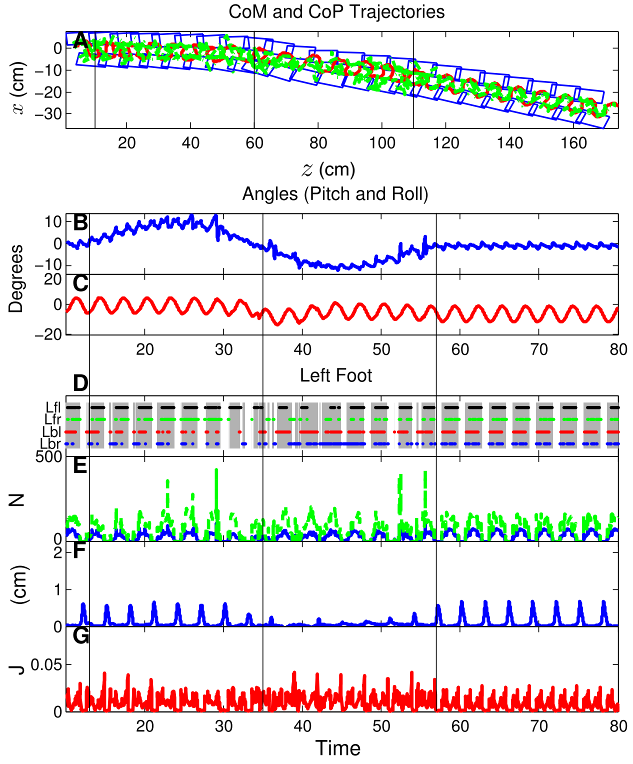


Fig. 21. Similar to Fig. 6, but for *Best controller* 2 solution with feedback inclusion in the full slope scenario. Vertical Lines indicate the moments the robot starts to climb, reaches the top and the bottom of the slope, respectively from left to right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

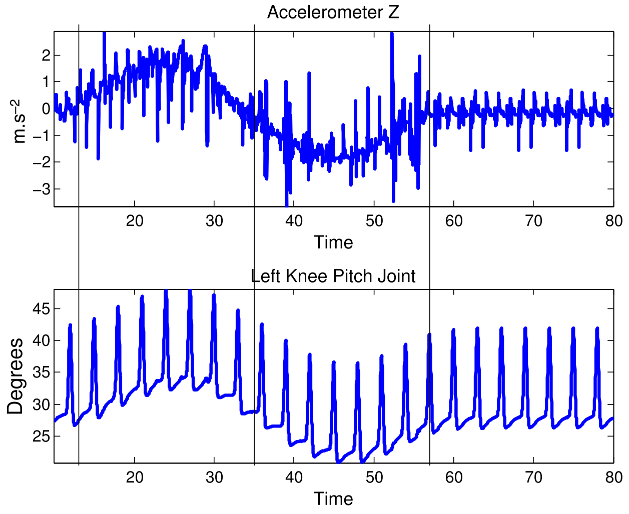


Fig. 22. Time evolution of the Z axis of the robot accelerometer (top) and the left knee pitch joint (bottom) during the full slope scenario, with *Best controller* 2 solution. Vertical Lines indicate the moments the robot starts to climb, reaches the top and the bottom of the slope, respectively from left to right.

Table 6
Feedback loops involved in the different scenarios of *Best controller* 2 solutions.

Scenarios	a_x (m s^{-2})	a_y (m s^{-2})	a_z (m s^{-2})	g_x (rad s^{-1})	g_y (rad s^{-1})	g_z (rad s^{-1})
Flat		y	y	y	y	y
Up	y	y	y			
Down		y				
Stage		y				

6. Discussion

In case of flat ground, the use of feedback pathways showed no specific advantage, since the solutions were very similar.

Further, considering other metrics for the performance of the generated movement, such as the movement of the CoM and CoP, pitch and roll angles, stance duration and vertical clearance, one can see that robot motion seems to be more entrained with the environment and the robot model when in closed-loop.

Evolution results in the slope scenarios demonstrated the ability to generate solutions capable to climb and descend slopes, both with and without the inclusion of feedback inputs. However, the achieved solutions could only perform in the tasks for which they were evolved to. Otherwise, the robot fell. For instance, the solutions that were evolved to climb the slope would fall if tried to descend that same slope.

In order to prevent this from happening, a different scenario composed of up and down stages was evaluated, in which each solution evolved in both stages. The results showed that only the solutions with feedback inclusion achieved adaptation to the environment in both stages. When no feedback was considered the robot fell during the climbing stage. Therefore, the use of feedback was relevant in order to enable adaptation to the ground's slope.

More important was to verify the impact that sensory information inclusion brings to the robot performance and how it enhances the locomotion skills of the biped robot. This was assessed through a functional gait analysis considering CoM and CoP trajectories, pitch and roll angles, stance duration and vertical clearance, as well as energy consumption levels. The resultant CoM trajectory was enlarged and smoothly oscillated between the center of both feet. Thus, the generated locomotion was more stable and thus entrained with the robot dynamics and involving environment.

On the overall the inclusion of feedback provides an advantage to the locomotion controllers. It enabled the creation of more general locomotion controllers, capable of adapting the locomotion to contrary situations, as well as enabled the generation of more efficient locomotion controllers in some scenarios (flat and up slope).

7. Conclusions

In this paper, we have proposed a gait optimization system for a biped robot using GP. Further, we explored the inclusion of feedback pathways to achieve locomotion adaptation to the environment. We based ourselves in previous work [5], in which a CPG based solution using a combination of a small set of motion primitives was able to generate biped walking for the DARwIn-OP humanoid robot.

Two controllers were proposed. *Controller* 1 generates biped locomotion for a target platform, and evolves in open-loop without including any sensory information from the environment. *Controller* 2 generates biped locomotion for the same target platform but in a closed-loop fashion. Therefore, sensory information was directly included into the movement generation in order to achieve adaptability to the environment.

The obtained results have shown the adequacy of both controllers to produce different walking locomotion with improved performance over the initial hand-tuned one, according to the

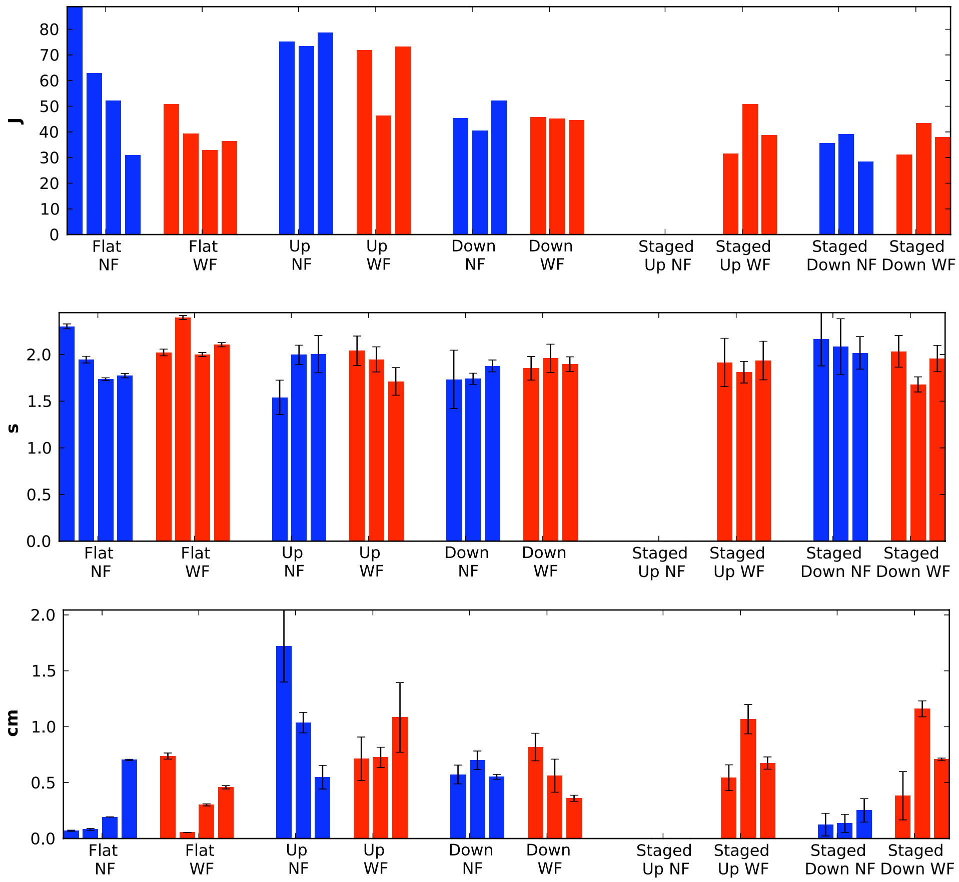


Fig. 23. Energy consumption levels (top) and Mean ± SD stance durations (middle) and ground clearance (bottom) values for the results of all evolution results.

specified criterion. In fact, the achieved displacement was up to four times larger. Further, the inclusion of sensory feedback pathways provided the required adaptability to achieve locomotion with better performance when climbing and descending slopes. These solutions were tested against the disabling of the feedback (replaced by a null value), and against scenarios not used in the evolution stages, demonstrating the generalization of the solution. The solutions were able to continuously adapt to the changing inclination of a complete sine curve like slope, and also were able to walk over slopes of lower inclination levels.

A functional analysis based on some metrics of the performed gait was done. This includes CoM and CoP trajectories, roll and pitch angles, stance duration and the achieved vertical clearance. Overall, the obtained results emphasize the fact that the inclusion of feedback enabled a smoother locomotion, with less undesired oscillations. The obtained locomotion was more entrained with the robot dynamics and the environment. Future work includes comparisons with other state of the art methods of optimization such as the Non-Dominated Sort Genetic Algorithm—NSGA II [15] and learning strategies such as Reinforcement Learning techniques [38,39]. Further we also want to quantify how the proposed framework managed to enhance the locomotion skills of the biped robot. In order to improve the efficiency of the evolutionary search, the application of genetic operators that explore the existence of correlations between the actuators will also be studied.

Additionally, experiments were performed in a physical platform. However, a great difference in the physical setup caused the robot to fall. Such problem is often referred to as Reality Gap. As future work, this problem will be addressed similarly to the proposed approach in [40], so as to provide an efficient locomotion controller for a real DarwinOP robot.

References

[1] Miomir Vukobratović, Branislav Borovac, Zero-moment point—thirty five years of its life, Int. J. Humanoid Rob. 1 (01) (2004) 157–173.

[2] Koichi Nishiwaki, Satoshi Kagami, Simultaneous planning of CoM and ZMP based on the preview control method for online walking control, in: 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), IEEE, 2011, pp. 745–751.

[3] Auke Jan Ijspeert, 2008 special issue: central pattern generators for locomotion control in animals and robots: a review, Neural Netw. 21 (4) (2008) 642–653.

[4] Hamed Shahbazi, Kamal Jamshidi, Amir Hasan Monadjemi, Sensor-based programming of central pattern generators in humanoid robots, Int. J. Adv. Robot. Syst. 10 (2013).

[5] V. Matos, Cristina P. Santos, Central pattern generators with phase regulation for the control of humanoid locomotion, Business Innovation Center Osaka, Japan, 2012.

[6] Gauss Lee, Robert Lowe, Tom Ziemke, Modelling early infant walking: testing a generic cpg architecture on the nao humanoid, in: 2011 IEEE International Conference on Development and Learning, ICDL, volume 2, IEEE, 2011, pp. 1–6.

[7] Martin Friedmann, Jutta Kiener, Sebastian Petters, Dirk Thomas, Darmstadt dribblers team description for humanoid kidsize league of robocup 2007, 2007.

[8] Jiang Shan, Fumio Nagashima, Neural locomotion controller design and implementation for humanoid robot HOAP-1, in: 20th Annual Conference of the Robotics Society of Japan, 2002.

[9] Youngwoo Kim, Yusuke Tagawa, Goro Obinata, Kazunori Hase, Robust control of CPG-based 3D neuromusculoskeletal walking model, Biol. Cybernet. (2011) 1–14.

[10] S. Aoi, N. Ogihara, T. Funato, Y. Sugimoto, K. Tsuchiya, Evaluating functional roles of phase resetting in generation of adaptive human bipedal walking with a physiologically based model of the spinal pattern generator, Biol. Cybernet. 102 (2010) 373–387.

[11] J. Morimoto, G. Endo, J. Nakanishi, S.H. Hyon, G. Cheng, D. Bentivegna, C.G. Atkeson, Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking, in: IEEE International Conference on Robotics and Automation, May 2006.

[12] Jimmy Or, A hybrid cpg–zmp controller for the real-time balance of a simulated flexible spine humanoid robot, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 39 (5) (2009) 547–561.

[13] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, M. Kawato, Learning from demonstration and adaptation of biped locomotion, Robot. Auton. Syst. 47 (2–3) (2004) 79–91.

[14] Jeong-Jung Kim, Jun-Woo Lee, Ju-Jang Lee, Central pattern generator parameter search for a biped walking robot using nonparametric estimation based particle swarm optimization, Int. J. Control Autom. Syst. 7 (3) (2009) 447–457.

[15] Miguel Oliveira, Vítor Matos, Cristina P. Santos, Lino Costa, Multi-objective parameter CPG optimization for gait generation of a biped robot, in: 2013 IEEE International Conference on Robotics and Automation, ICRA, Karlsruhe, May 6–10, 2013, 2013.

[16] Naofumi Miura, Yu Ikemoto, Jose Gonzalez, Jun Inoue, Wenwei Yu, Analyzing compensation strategy in impaired walking using a humanoid robot, Trans. Control Mech. Syst. 1 (1) (2012).

[17] Krister Wolff, Generation and Optimization of Motor Behaviors in Real and Simulated Robots, Chalmers University of Technology, 2006.

[18] Kazuo Miyashita, Sooyol Ok, Kazunori Hase, Evolutionary generation of human-like bipedal locomotion, Mechatronics 13 (8) (2003) 791–807.

[19] Norikazu Sugimoto, J. Morimoto, Phase-dependent trajectory optimization for CPG-based biped walking using path integral reinforcement learning, in: 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids), IEEE, 2011, pp. 255–260.

[20] Gen Endo, Jun Morimoto, Takamitsu Matsubara, Jun Nakanishi, Gordon Cheng, Learning CPG-based biped locomotion with a policy gradient method: application to a humanoid robot, Int. J. Robot. Res. 27 (2) (2008) 213–228.

[21] Hamed Shahbazi, Kamal Jamshidi, Amir Hasan Monadjemi, Modeling of mesencephalic locomotor region for nao humanoid robot, Ind. Robot Int. J. 39 (2) (2012) 136–145.

[22] Joseph H. Solomon, Mark A. Locascio, Mitra J.Z. Hartmann, Linear reactive control for efficient 2D and 3D bipedal walking over rough terrain, Adapt. Behav. 21 (1) (2013) 29–46.

[23] Saunders, Verne T. Inman, Howard D. Eberhart, The major determinants in normal and pathological gait, J. Bone Joint Surg. Am. 35 (3) (1953) 543–558.

[24] Yvan Bourquin, Auke Jan Ijspeert, Inman Harvey, Self-organization of locomotion in modular robots, Unpublished Diploma Thesis, 2004. http://birg.epfl.ch/page53073.html.

[25] Auke Jan Ijspeert, Alessandro Crespi, Jean-Marie Cabelguen, Simulation and robotics studies of salamander locomotion, Neuroinformatics 3 (3) (2005) 171–195.

[26] Mikhail Prokopenko, Vadim Gerasimov, Ivan Tanev, Measuring spatiotemporal coordination in a modular robotic system, in: Artificial Life X: Proceedings of The 10th International Conference on the Simulation and Synthesis of Living Systems, Bloomington IN, USA, 2006, pp. 185–191.

[27] L. Gritz, J.K. Hahn, Genetic programming evolution of controllers for 3-D character animation, in: Genetic Programming, vol. 97, 1997.

[28] I. Tanev, T. Ray, A. Buller, Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot, IEEE Trans. Robot. 21 (4) (2005) 632–645.

[29] B. Andersson, P. Svensson, M. Nordahl, P. Nordin, On-line evolution of control for a four-legged robot using genetic programming, in: Real-World Applications of Evolutionary Computing, 2000, pp. 322–329.

[30] K. Wolff, M. Wahde, Evolution of biped locomotion using linear genetic programming, in: Climbing and Walking Robots, Towards New Applications, 2007.

[31] K. Wolff, P. Nordin, Learning biped locomotion from first principles on a simulated humanoid robot using linear genetic programming, in: Genetic and Evolutionary Computation—GECCO 2003, SpringerVerlag, 2003, pp. 12–16.

[32] S. Ok, K. Miyashita, K. Hase, Evolving bipedal locomotion with genetic programming—a preliminary report, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, Volume 2, IEEE, 2001, pp. 1025–1032.

[33] S. Ok, D.S. Kim, Evolution of the CPG with sensory feedback for bipedal locomotion, in: Advances in Natural Computation, in: Lecture Notes in Computer Science, vol. 3611, 2005, pp. 714–726.

[34] P.E. McSharry, G.D. Clifford, L. Tarassenko, L.A. Smith, A dynamical model for generating synthetic electrocardiogram signals, IEEE Trans. Biomed. Eng. 50 (3) (2003) 289–294.

[35] Auke J. Ijspeert, Jun Nakanishi, Stefan Schaal, Learning rhythmic movements by demonstration using nonlinear oscillators, in: Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems, IROS2002, vol. 2002, 2002, pp. 958–963.

[36] Christian Gagné, Marc Parizeau, Open beagle: a C++ framework for your favorite evolutionary algorithm, SIGEVOlution 1 (2006) 12–15.

[37] O. Michel, Webots: professional mobile robot simulation, J. Adv. Robot. Syst. 1 (1) (2004) 39–42.

[38] Marc Peter Deisenroth, Roberto Calandra, André Seyfarth, Jan Peters, Toward fast policy search for learning legged locomotion, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2012, pp. 1787–1792.

[39] Jun Morimoto, Jun Nakanishi, Gen Endo, Gordon Cheng, Christohper G. Atkeson, Garth Zeglin, Poincare-map-based reinforcement learning for biped walking, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, ICRA 2005, IEEE, 2005, pp. 2381–2386.

[40] Sylvain Koos, Jean-Baptiste Mouret, Stéphane Doncieux, Crossing the reality gap in evolutionary robotics by promoting transferable controllers, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO'10, ACM, New York, NY, USA, 2010, pp. 119–126.

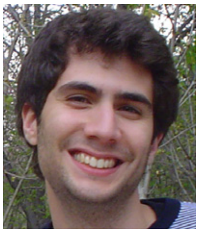
Pedro Silva received the B.S. and its M.Sc. degrees in Informatics from the Informatics Department In University of Minho, Braga, Portugal, in 2008 and 2010, respectively.

He has worked during its M.Sc. and after that as a researcher in the Adaptive Systems Behavior Group in the Industrial Electronics Department of the University of Minho, Guimarães, Portugal. His main interests are learning and adaptation in robotics, specially in legged robots.

Cristina P. Santos received the B.S. degree in Industrial Electronics, the M.Sc. degree in Robotics, and the Ph.D. degree in Robotics in the field of Nonlinear dynamics, all from University of Minho, Guimarães, Portugal, in 1994, 1998 and 2003 respectively. There she works as an Auxiliary Professor since 1996. The Ph.D. was also in collaboration with the CNRS–CNRC Marseille, France.

Her research focuses on the extension of the use of the dynamical systems theory to the achievement locomotion for multi-dof robots; achieve cooperation among multi-robots and learning; characterize human motion, and designing robots and robot controllers for rehabilitation of patients suffering from motor problems.

She has authored/co-authored more than 40 research papers, in international journals and conferences.



Vítor Matos is a Ph.D. student in Robotics at University of Minho. He holds a M.Sc. degree in Industrial Electronics from University of Minho since 2009. His current research includes robot locomotion and dynamical systems theory for achieving robust generation of rhythmic and discrete motions.

Lino Costa received the B.S. degree in Systems and Informatics Engineering, the M.Sc degree in Informatics, and the Ph.D. degree in Production and Systems Engineering in the field of Multiobjective and Nonlinear Optimization, all from University of Minho, Braga, Portugal, in 1993, 1996 and 2003 respectively. He works as an Assistant Professor in the Department of Production and Systems of the School of Engineering in the University of Minho.

His research focuses on applied statistics, nonlinear and multiobjective optimization and evolutionary algorithms. He has authored/co-authored more than 40 research papers, in international journals and conferences.