

Evolution of Neuro-Controllers for Trajectory Planning Applied to a Bipedal Walking Robot with a Tail

Álvaro Gutiérrez¹, Fernando J. Berenguer² and Félix Monasterio-Huelin¹

¹*Universidad Politécnica de Madrid*

²*Fundación PRODINTEC*
Spain

1. Introduction

Bipeds are complex hybrid dynamical systems, in the sense that they mix both continuous and discrete-event phenomena (Hurmuzlu et al., 2004). The main characteristic of biped walkers is the abrupt kinematics change between the aerial phase and the support phase accompanied of dynamical impacts. The main problem is how to achieve a rhythmical or periodical walk. One of the main problems of these robots is their high power and energy consumption, which limits mainly their autonomy. It could be attributed to, for example, the high number of actuated joints (about 20), and also because the study of energy consumption is not often considered during the planning of movements.

The construction of a locomotion controller for completely actuated legged robots is performed in many forms. Firstly by obtaining predetermined trajectories, using techniques based on kinematical models or Artificial Intelligence methods like the evolution of locomotion based on fuzzy controllers (Magdalena & Monasterio, 1995) or spline controllers (Boeing & Bräunl, 2004) for example. These studies are oriented to solve trajectory generation problems for the active control centred approach. In (Löffler et al., 2004) there is an example of this two-part methodology showing that an impedance control is better than the computed torque method. Despite this better performance, the second algorithm needs both a position sensor and a velocity sensor, and the impedance controller needs also force-torque sensors.

The conventional approach has been questioned by researchers inspired by biomechanical models (McMahon, 1985 and Kuo, 2007). The discovery by McGeer (McGeer, 1990) of passive dynamic walking by building a biped without any motors or sensors opens the doors to a new design concept based on morphological considerations. The interaction or trade-off between morphology and control is in the centre of the more recent research and debates in robotics (Matsushita et al., 2006 and Pfeifer & Bongard, 2007). Exploiting the intrinsic passive dynamics has many advantages against the classical two-part methodology (trajectory generation plus active controller). Two of them are the energy consumption reduction (the only energetic cost occurs in step-to-step transitions) and the control simplicity (low computational cost).

Nevertheless, some theoretical and practical studies (Collins & Ruina, 2005) show that it is difficult to achieve the complex dynamic exhibited by the human and animal locomotion taking into account only the properties of the simple passive-dynamic walking, but might help to design walker robots. How to exploit the above-mentioned passive properties of biped robots with the incorporation of sensors is studied in (Iida & Pfeifer, 2006). Other studies make use of passive trajectories to design active controls under the classical methodology. Some researches are based on kinematics (Asano et al., 2004) and other on energy constraints (Asano et al., 2005).

Conventional models of bipedal walking generally assume rigid body structures, while elastic material properties seem to play an essential role in nature (Alexander, 2008). Spring-damper elements have been proposed to advance one solution under the compliant leg concept (Iida et al, 2008, Geyer & Seyfarth, 2006 and Iida et al., 2007) or through an adaptable compliance (inverse of stiffness) mechanism (Van Ham et al. 2007).

Passive biped robots have stable limit cycles when they go down a small slope (3° - 5°). In (Siqueira & Terra, 2006) a torso was added to a passive biped robot. Due to this fact a stable limit cycle cannot be found. Then the authors add two PD controllers to keep the torso in a fixed position to achieve the walk. In (Berenguer & Monasterio-Huelin, 2006, 2007a, 2007b, 2007c and 2008) an actuated tail is added to produce the walk. In both cases a spring is added to the ankle to transform the stored potential energy into kinematics one.

In this work we use the last model to study the problem of going up a small slope (0° - 10°). To achieve this objective it is necessary to solve a trajectory planning and a control problem. As has been said, classical methods solve both problems separately adding to the robot one actuator (motor) for each degree of freedom. This technical solution could be avoided taking into account a different philosophy. Instead of generating an explicit desired path defined as a function of kinematics variables (positions and velocities) or even dynamic variables (forces and torques) we can design the robot having the ability to solve the task by changing in real-time some mechanical parameters in function of the environment variations (in our case the slope of the ground). The path will then be generated implicitly by exploiting its intrinsic dynamic. Following this philosophy we can say that modifying some local characteristics of the robot it is possible to achieve a global desired behaviour adapted to the environment.

In this work we investigate the use of dynamical recurrent neural networks to vary in real-time the parameters of the springs of the ankles, and also the frequency of oscillation of the tail, without calculating the exact trajectory the robot must follow. This class of neural networks (Beer & Gallagher, 1992), consists on a coupled set of first order non-linear differential equations whose parameters must be tuned. To do so we use an off-line optimization method through genetic algorithms (Holland, 1975 and Goldberg, 1989). This so-called evolutionary methodology (Nolfi & Floreano, 2000) will be described in Section 4.

In the rest of this chapter, we present the details of our solution and analyse its performance. Robot description is shown in Section 2. Section 3 relates the objective of the framework and the study on the dynamic variation of the robot's parameter. In Section 4, the evolutionary algorithm proposed is explained. Robot implementation on the simulator is shown in Section 5. Results are shown in Section 6. Finally the conclusions can be found in Section 7.

2. Robot Description

2.1 Kinematical description

The system studied in this chapter is a mechanism with the morphology of a biped with a tail (with ankles and without knees) which is able to walk using just one actuator. Each leg and foot forms a four link parallel kinematics chain with only one passive degree of freedom when the flat rectangular foot is attached to the ground. The ankle has a spring which will be described later as it is the main element of the present study. The mechanical structure of the system is shown in Figure 1. Both legs are joined by a body/torso through two revolute joints. The torso mechanism is a new parallel kinematics chain included into the system to maintain both legs parallel to the frontal view (see Figure 2). The tail is attached to the torso through the only actuated joint.

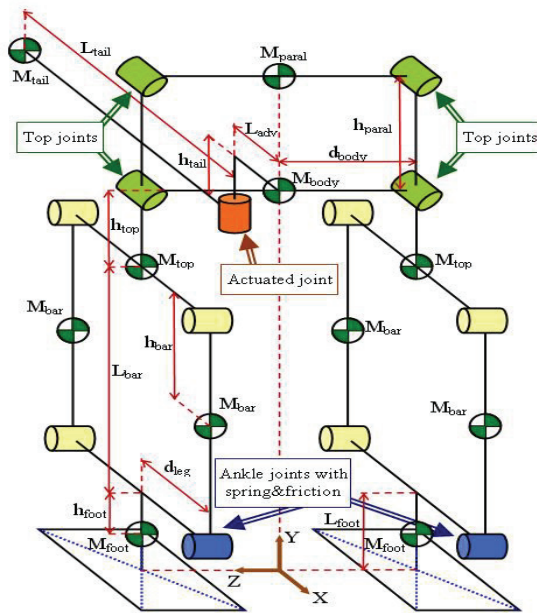


Fig. 1. Model of the biped mechanism.

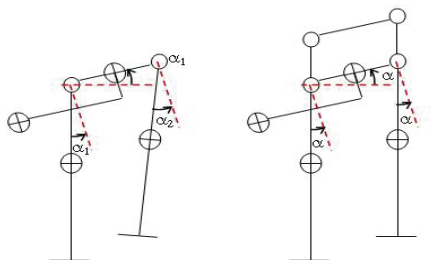


Fig. 2. Torso parallel kinematics chain

Due to the characteristics of the mechanism the model has thirteen joints with only four degrees of freedom, three passive and one actuated. When both feet are on the ground the parallel mechanism of the torso is redundant. The feet contact point with the ground adds two new passive joints. Nevertheless the dimensions of the feet and the model selected for the eight point contact forces with the ground allows us not to consider them from a static point of view. The mechanism remains standing if the springs of the ankles are properly selected.

Some of the important characteristics that this mechanism provides are the following ones:

- The springs in the ankle joints provides the needed force to keep the robot in a stable position, in the case it is not actuated (e.g. switched off). In this state the robot does not need energy consumption.
- When the tail is designed following the condition presented in (Berenguer & Monasterio-Huelin, 2007b), it acts as a counterbalance where the length of a single support phase is no limited in time. It allows the system to remain with a foot raised during an indefinite time. Then, the turn of the mechanism can be reached during a single support phase by adding new joints in the feet or in the hip of the mechanism.
- When all the kinematics parameters have been defined, the behaviour depends only on the tail trajectory and the ankle parameters (friction, stiffness and equilibrium position). We suppose the friction negligible for the rest of joints. In this study we consider the ankle friction (b), the stiffness (K) and the equilibrium position (θ_0) variable parameters. In a real robot we can modify these parameters using a system like MACEPPA (Van Ham et al. 2007) for example.
- If θ_0 is zero, the biped also walks thanks to the force exerted by the tail over the body when the tail is in a lateral position (Berenguer & Monasterio-Huelin, 2007c). Normally the system walks by a combination of this force and the gravity effect.
- Finally, as we showed in (Berenguer & Monasterio-Huelin, 2007b), the robot parameters can be adjusted to obtain a smooth ground contact at the end of each single support phase. This is an interesting characteristic if we try to reduce the dissipated energy during its walk.

The main parameters of the mechanism used on simulations are presented in Table 1. Simulations have been made using MATLABTM and the SimMechanics toolbox.

Model parameters for simulations					
Name	Value	Name	Value	Name	Value
M_{body}	50.0gr	L_{adv}	0.0mm	H_{paral}	50.0mm
M_{top}	50.0gr	L_{bar}	400.0mm	h_{top}	30.0mm
M_{bar}	200.0gr	L_{foot}	10.0mm	h_{bar}	200.0mm
M_{foot}	150.0gr	L_{tail}	150.0mm	h_{foot}	5.0mm
M_{leg}	450.0gr	d_{body}	50.0mm	h_{tail}	15.0mm
M_{tail}	700.0gr			d_{leg}	20.0mm
M_{paral}	5.0gr			H_{T}	510.0mm
M_{T}	1955.0gr				

Table 1. Parameters for simulations.

2.2 Gait description

The tail of the robot moves in an almost horizontal plane. When the tail is in a lateral position of the mechanism, its mass acts as a counterbalance and produces the rise of one of the feet, where a step movement begins. We will define and describe here seven phases during a stride. Figure 3 shows these phases starting at an equilibrium position of the system with the tail in its central position.

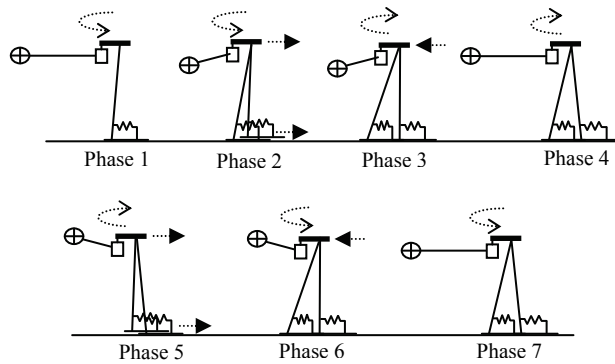


Fig. 3. Phase during a stride

- *Phase 1: Displacement of the tail to a lateral of the mechanism:* Both springs hold the weight of the mechanism, and this one stays almost vertical. We use linear springs in Figure 3 for a better understanding of their effect and because they have been used in the construction of our first real prototype Zappa.
- *Phase 2: Rise of one foot and single support phase:* When a foot rises, only one spring holds the body, so the stance leg falls forward to a new equilibrium position. In this phase, kinetic and potential energies are transformed into elastic energy and stored in the ankle springs. The swing leg moves forward as a pendulum.
- *Phase 3: Contact of the swing leg with the ground:* At this moment the greater kinetic energy losses are due to the collision. The designer must calibrate the mechanism trying to reduce the velocities at this moment and provide a smooth contact between the foot and the ground.
- *Phase 4: Movement of the tail to the other side:* In this double support phase, the projection of the centre of mass of the mechanism moves from one foot to the other. The body moves backwards to a position in which both springs generate opposite torques.
- *Phase 5: Rise of the second foot:* In this phase, the spring of the foot that is in the ground produces enough torque to take the body forward again.
- *Phase 6: New contact of a swing leg with the ground:* Same as phase 3.
- *Phase 7: New displacement of the tail during a double support phase:* If a new stride is desired, this phase represents returning to phase 1. If the tail stops in the middle position, the system will stay in a steady configuration with no energy cost.

3. The objective framework

This paper is centred on the study of the dynamic variation of the parameters of the torsional spring situated at the ankles and the frequency of the tail oscillation. Previous investigations show that these parameters allow the robot to change the step length and the velocity of the walking (Berenguer & Monasterio-Huelin, 2008). To get this dynamic variation, a torque is applied to each ankle following a function given by equation 1.

$$\tau_{ankle} = -K(q - \theta_0) - b\dot{q} \quad (1)$$

The three time varying parameters (K , θ_0 , and b) are the same for both ankles. Below we explain this time variation. The spring-actuator is compressed and stores some part of the dynamic energy. At the stand situation this torque does not move the robot but changes the initial angle between the legs and the foot.

The tail is actuated through a PD controller that follows a sine reference signal,

$$q_{tail} = \frac{\pi}{2} \sin(2\pi ft) \quad (2)$$

The frequency (f) is also a time-varying parameter of the model.

The objective of this paper is to make the robot to go up flat surfaces of different slopes. The hypothesis we envisaged is that the initial inclination or body posture is crucial to get this objective. This hypothesis is bio-inspired because the adaptation of postural orientation to changes in surface inclination is a known reflex behaviour of many animals including humans (Prentice et al., 2004, Kluzik et al., 2007b and Edwards, 2007). It is also known that there exist postural after-effects of stepping on an inclined surface. This effect could be due not to a local adaptive mechanism (i.e. changes in ankle angles) but to a central adaptive mechanism that affects the postural geometry more globally (Kluzik et al., 2007a). In this work we do not consider completely this phenomenon, but because we change local parameters in function of an environment variable (the ground slope) we are not contradicting these discoveries.

Some simulation studies described later show that the hypothesis is true when the robot begins to walk on a constant slope flat surface. But, how to select the parameters when the slope of the ground varies while the robot is in movement? We have decided to add to the robot a simple continuous time recurrent neural network to obtain the time variation of the three parameters of the torque applied to the ankles and the frequency of the tail oscillation (K , θ_0 , b , f).

This approach is different from others explained previously in the sense that the postural phenomenon is here implemented by adjusting mechanical parameters in function of the ground slope. On the contrary, in (Maurer & Peterka, 2005), a two degrees of freedom PID controller (Aström & Hägglund, 1995) is tuned to follow a constant reference signal based on the ground slope. The output of the controller is the torque applied to the ankle.

We have simplified the input to the neural network to only one variable, the ground's slope. To move these ideas into a real robot it is necessary not only to add a mechanism that would

be able to change dynamically the parameters of the ankle, but a sensor that can measure the slope of the ground (a perceptual sensor). In real studies with the robot Zappa (Berenguer and Monasterio-Huelin, 2008) this measurement has been made with accelerometers sensors showing that it is possible to estimate the inclination of the legs on a flat surface with a null slope. Figure 4 shows the accelerometers location and their sensing directions over the model in a sagittal view. (The accelerometers measure $a-g$, where a is the translational acceleration vector of the body and g the gravity vector.)

Due to the kinematical design of the mechanism the reference coordinate system attached to the torso always has the X axis (the direction of the robot advance) parallel to the ground. As a consequence it is possible to get a realistic measure of the ground slope putting an accelerometer on the torso

4. Evolutionary Robotics

Evolutionary Robotics is a methodological tool to automate the design of robots' controllers based on the use of artificial evolution to find sets of parameters for artificial neural networks that guide the robots to the accomplishment of their task (Nolfi & Floreano, 2000). In most cases, the control systems of evolutionary robots are artificial neural networks evolved through genetic algorithms, explained in the next sections.

4.1 Genetic Algorithms

Genetic algorithms are a general-purpose search algorithms inspired by natural population genetics to evolve solutions to problems (Holland, 1975). A genetic algorithm operates on a population of artificial chromosomes by selectively reproducing the chromosomes of individuals with higher performance and applying random changes. Each structure in the population represents a candidate solution to the problem and is ranked according to a *fitness* function. This ordered list is used to create new individuals and repeat the process until the optimization problem is satisfied.

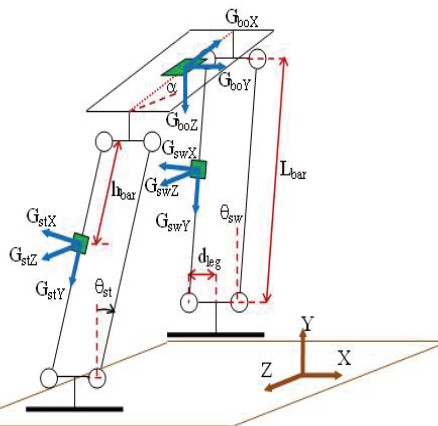


Fig. 4. Accelerometers location and their sensing directions over the model in sagittal view (Tail and frontal parallel link mechanism are not shown).

A genetic algorithm starts with a fix size population of randomly generated individuals and advance towards better solutions by applying genetic operators. Each individual is coded in a chromosome which might implied several variables such as the connection weights of a neural network, gains to apply to a specific input or a time-delay constant for an output. A fitness function must be devised for each problem to be solved.

Given a particular chromosome, the fitness function returns a single numerical fitness which is supposed to be proportional to the adaptation of the individual that the chromosome represents. Once a generation of individuals has been evaluated, selective reproduction starts to create new copies of the best chromosomes in the population. Individuals with higher fitness values tend to leave a higher number of copies for the next generation. Given an individual x_i and its fitness f_i , chromosome reproduction probability p_i is denoted by Equation 3.

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}, i = 1 \dots n \quad (3)$$

Commonly, each generation has the same number of individuals. The process which determines how the new generation is created depends on many factors. Firstly, the selection algorithm must be chosen. The new generation can be obtained from the last generation whole population, or choosing a given number of best individuals which will create copies of themselves. Some of the best individuals (*Elites*) could also be inherited directly by the next generation.

After selection has been carried out, the construction of the new generation is completed by recombination and mutation. Recombination is achieved by the *crossover* operator, which combines the chromosomes of two parents. Parents are chosen randomly from the new generation and a random point is selected around which the genome is swapped between the two individuals (Figure 4a). The *mutation* operator arbitrarily alters one or more genes of a selected chromosome (Figure 4b).

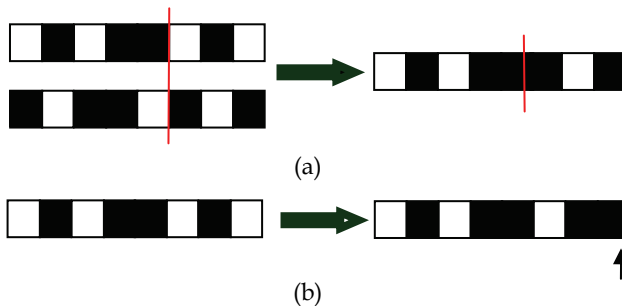


Fig. 5 (a) Crossover: Given two selected genotypes, genetic material is swapped around a random selected point. (b) Mutation: Given one genotype, a random selected gene is mutated. Once the new generation has been created a new evaluation of all individuals

occurs, and the process is repeated for several generations during which the average and fitness values are monitored.

The basics of this simple genetic algorithm and several variations are well described in many previous books (Goldberg, 1989)

4.2 Neural Network

The scientific study of the nervous system led to the need to obtain mathematical models of the neurons. The first model (McCulloch & Pitts, 1943) used a two-state threshold system that followed a stochastic algorithm involving sudden changes of its state at random times. Later (Hopfield, 1984) a deterministic continuous-time mathematical model was proposed for better approaching to the functioning of the real neurons.

Neural networks are interconnected neurons that have many interesting properties (Amit, 1989 and Rumelhart & McClelland, 1986).

In this work we use the continuous-time recurrent neural network (CTRNN) (Beer & Gallagher, 1992) that obeys the following set of non-linear first-order differential equations:

$$\tau \dot{y} = -y + W\sigma(y + \theta) + I \quad (4)$$

where y , I , σ and θ are $[n \times 1]$ vectors and W is an $[n \times n]$ matrix, where y_i is the state variable of the i th neuron (the mean membrane potential); w_{ij} , $j \neq i$ represent synaptic connections from neuron j to neuron i ; the self-interaction w_{ii} represents a simple active conductance; θ_j is a constant threshold or bias; τ_i is a time constant (resistance x capacitance); I_i the external input, and σ_j is a sigmoid function (the mean firing rate of the neuron):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

CTRNN's are a special case of the general class of additive neural network models (Grossberg, 1988) which have a complex dynamical behaviour. In (Beer, 2006) the parameter space structure of the CTRNN is systematically studied.

From a mathematical point of view the input to the system is $(I_i + \theta_i)$ since equation 4 can be rewritten in the form

$$\tau \dot{z} = -z + W\sigma(z) + I + \theta \quad (6)$$

using the transformation $z_i = y_i + \theta_i$.

With the transformation $y = Wx + I$ (and supposing W is not a singular matrix) we obtain another useful general representation for analytical studies (Vidyasagar, 1993):

$$\tau \dot{x} = -x + \sigma(Wx + I + \theta) \quad (7)$$

These networks are perfectly suitable for evolutionary robotics tasks due to many characteristics:

- The network is composed of a large number of highly interconnected processing neurons working in parallel to solve a specific problem.
- Neural networks have the ability to learn from experience (that is, from historical data collected in some problem domain).
- Sensors and actuators can be mapped directly into the inputs and outputs neurons.
- Smooth changes on the parameters will correspond to gradual changes of its behaviour.
- After a neural network has been tuned, it can be deployed within an application and used to make decisions or perform actions when new data is presented.
- The network is robust to noise. Due to the weighted sum of input signals, oscillations in individual values do not severely concern the behaviour of the network.

4.3 Implementation

Our robot is composed of a CTRNN of 2 inter-neurons and an arrangement of 1 input neuron and 4 output neurons. Continuously the network receives as input the ground slope. The inter-neuron network is fully connected. Additionally, each inter-neuron receives one incoming synapse from the input neuron. Each output neuron receives one incoming synapse from each inter-neuron (see Figure 6). There are no direct connections between input and output neurons. The states of the output neurons are used to set the frequency of the robot's tail (f), and the three parameters (K , θ_0 and b).

The neural network obeys the state equations 8, 9 and 11, and the output equation 12.

$$x(t) = \gamma(t)g \quad (8)$$

where g is the gain factor, and γ is the ground slope.

$$\tau_i \dot{y}_i(t) = -y_i(t) + \sum_{j=1}^2 w_{h_j h_i} \sigma(y_j(t) + \theta_j) + I_i(t), i = 1, 2, \quad (9)$$

where

$$I_i(t) = w_{I h_i} \sigma(x(t) + \theta_I), i = 1, 2 \quad (10)$$

$$o_i(t) = \sum_{j=1}^2 w_{h_j o_i} \sigma(y_j(t) + \theta_{h_j}), i = 1 \dots 4 \quad (11)$$

$$out_i(t) = \sigma(o_i(t) + \theta_{o_i}), i = 1 \dots 4 \quad (12)$$

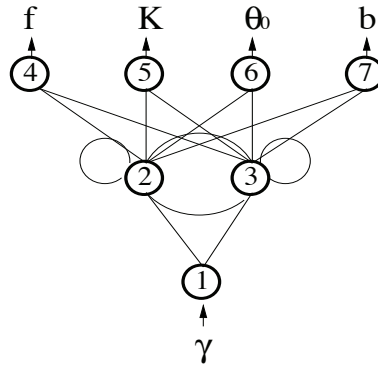


Fig. 6. Neural network architecture implemented. All the feedforward and recurrent connections are drawn (See the text for meaning of the labels).

A simple generational genetic algorithm is employed to set the parameters of the networks (Goldberg, 1989). The population contains 20 genotypes. Genotypes of the first generation are generated randomly. Generations following the first one are produced by a combination of selection with elitism, recombination and mutation. For each new generation, the five highest scoring individuals (“the elite”) from the previous generation are retained unchanged. The remainder of the new population is generated by fitness-proportional selection (also known as roulette wheel selection) from the individuals of the old population. Each genotype is a vector comprising 24 real values (i.e., 14 connection weights, 2 decay constants, 7 bias terms, and 1 gain factor). Initially, random genotypes are generated by initializing each of their components to values chosen at random by using a continuous uniform distribution in the $[0,1]$ interval. New genotypes, except “the elite”, are produced by applying recombination with a probability of 0.5, or if there has been no recombination, a mutation rule is applied with probability 1.

During evolution, all vector component values are constrained to remain within the range $[0,1]$. Genotype parameters are linearly mapped to produce network parameters with the following ranges: biases $\theta_i \in [-4, 4]$, weights $\omega_{ij} \in [-6, 6]$, gain factor $g \in [-5, 5]$ and decay constants $\tau_i \in [0.1, 100]$.

During evolution each genotype is cloned into the simulated robot and evaluated for a trial. Within a trial, the robot life-span is 25 simulated seconds. In each trial the robot is rewarded by an evaluation function F (see equation 15 below) which seeks to assess the ability of the robot to perform the task.

Once each population has been evaluated the different chromosomes are ranked according to the value obtained from the fitness function, and all the genetic operators are performed according to the rules described above.

5. Simulator

Robot is simulated thanks to the *SimMechanics* toolbox included in the *Matlab* software. Previous studies have depicted the simulation schema and only modifications on the previous structure are shown in this chapter. For further details see (Berenguer & Monasterio-Huelin, 2007b).

First modification is the mechanisms included for the on-line slope modification. In principle modifying the ground was a hard task due to the simulator architecture. The problem's solution was the inclusion of a mechanism to alter the gravity shown in Figure 7.

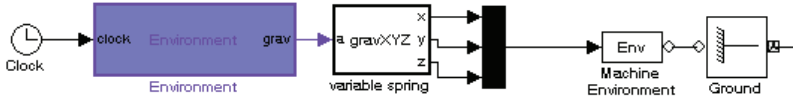


Fig. 7. Online gravity modification.

The gravity acts as a function of the real slope, where its 3 components (g_x, g_y, g_z) are defined in equation 13.

$$\begin{cases} g_x = -9.81 \sin(\gamma) \\ g_y = -9.81 \cos(\gamma) \\ g_z = 0 \end{cases} \quad (13)$$

Another enhancement implemented is the change of the previous static springs by the variable ones. As shown in Figure 8, this modification allows the robot to control its spring force. This spring is implemented following the Equation 1 explained in previous sections, where parameters q and dq/dt come from the ankle sensor and K, b and θ_0 are outputs from the neural network.

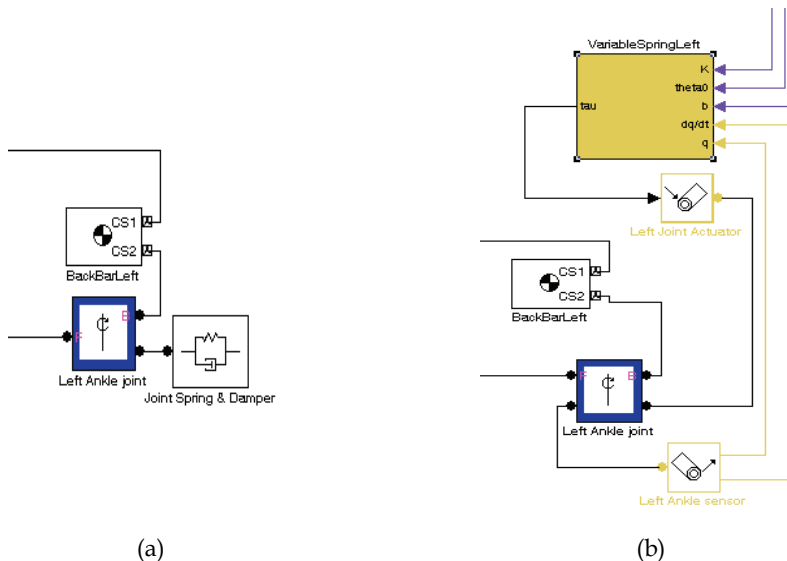


Fig. 8. (a) Spring implementation on previous works where the spring has constant parameters. (b) Variable spring implementation. The spring has time varying parameters.

Final modification is the inclusion of the neural network controller explained in Section 4.3. This CTRNN is in charge of getting the environment input (ground slope) and give the actuating parameters to the tail and variable springs. We observe that equation 9 can be rewritten as shown in Equation 14 and the block diagram is shown in Figure 9.

$$\tau_i \dot{y}_i(y) = -y_i(t) + \Delta_{h_i}, i = 1,2 \tag{14}$$

The task performs in the simulator for 25 simulated seconds. During this time the slope of the ground (translated into gravity changes) is increased each 5 seconds in 2°, starting in 0° and finishing in 8°. And external program is in charge of providing the genotype to the network and collecting the necessary output data. Once the trial has finished, the program stops the simulation, scores the individual and run next individual genotype. The individual is evaluated according to Equation 15, which measures the distance travelled by the robot during the experiment.

$$F = \sqrt{(x_f - x_0)^2 + (z_f - z_0)^2} \tag{15}$$

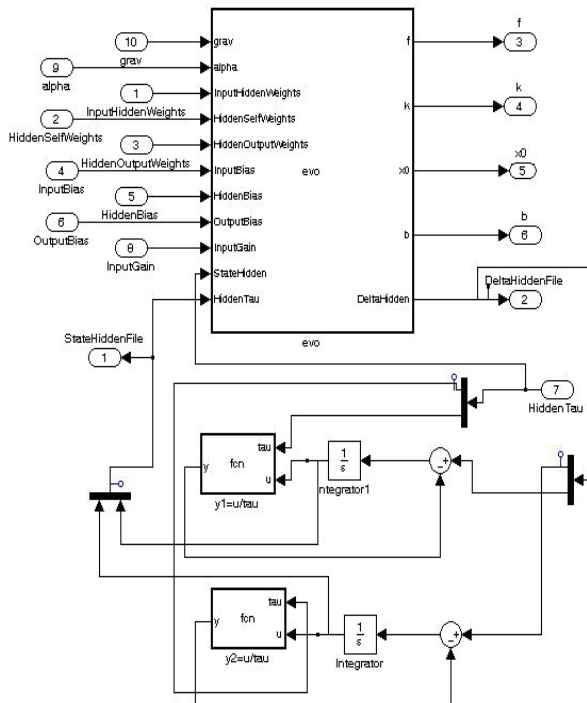


Fig. 9. CTRNN implemented on the simulator.

When a generation has been evaluated, the system rearranges the genotypes creating a new generation based on the elitism, mutation and crossover. This cycle is repeated for several evaluations until the robot reaches its travelled distance limits.

6. Results

Through a collection of patient simulation tests we have discovered that the robot can go up different constant slopes in the ranges shown in Table 2. This study has been made to accelerate the evolution process, but it has the handicap that must be applied to unnormalize the output of the neural network. This empirical study might be avoided in the future because it could not be necessary that the parameters change suddenly with the slope.

f	K	b	θ_0	slope (γ)
0.5-0.7	9-10	0.4-0.6	0.05-0.07	0°-6°
			0.07-0.12	6°-8°

Table 2. Output neural network parameters range.

We run simulations for 70 generations. As shown in Figure 10, the fitness reached a plateau around the value of 3.0 after generation 59.

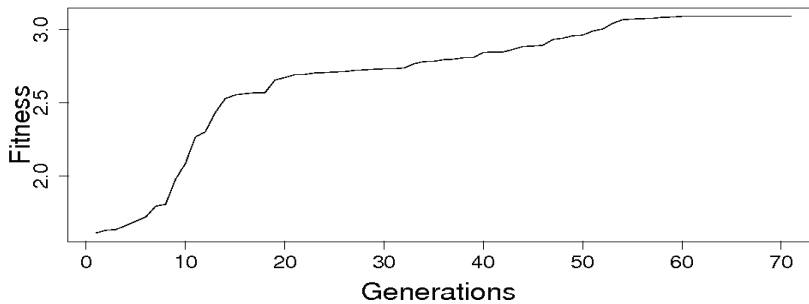


Fig. 10. Fitness of the robot at each generation.

The best performing genotype resulting from the evolutionary process was decoded again into the simulated robot to make measurements about the behaviour of the robot. We first studied the behaviour of the neural network, observing the performance of the hidden

neurons. Figure 11 shows the state values and delta values of both hidden neurons.

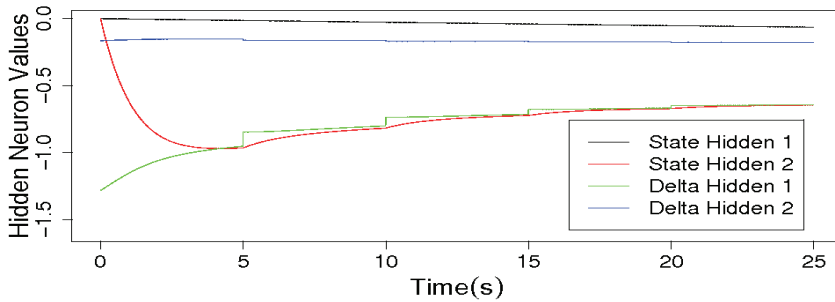
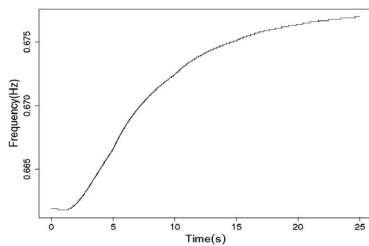


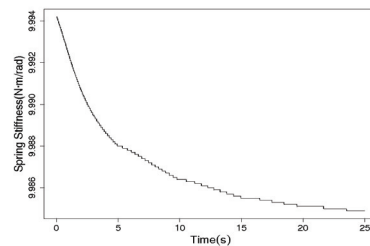
Fig. 11. Delta and State values for both hidden neurons in the best performing genotype.

As shown in the previous figures, hidden neuron delta values change abruptly with input changes (each 5 seconds) while the state values do it smoothly.

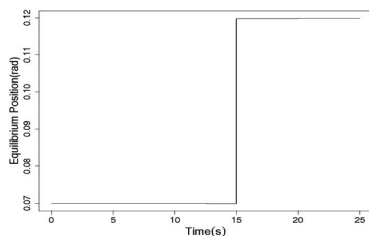
Output values for the neural network are shown in Figure 12. These values change over the time due to the environment changes. We observe that frequency is changing its value slowly following a transient state towards a final steady-state which is not able to reach. Stiffness (K) (figure 12b) is also following an effortlessly change, but adjusts on its inclination are found on the ground slope changes. Finally θ_0 (figure 12c) and b (figure 12d) modify their values abruptly, just one change for θ_0 matching one ground slope change ($t=15s.$) due to the output value range forced by design.



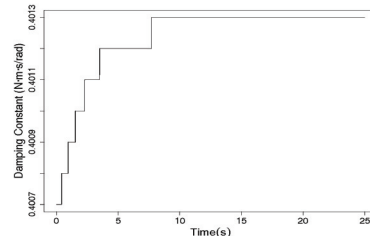
(a)



(b)



(c)



(d)

Fig. 12. (a) Tail frequency oscillation (f) (b) Spring stiffness (K) (c) Equilibrium position (θ_0) (d) Damping constant (b) values for the best genotype in a complete trial.

The spring values coming from the network go to the variable spring which sensing the ankles' angular position and velocity (Figure 13a) impinging a torque on both ankles (Figure 13b). We observe periodic oscillations on all the signals, due to the movement of the robot. While one foot is on the ground the other is moving forwards with some small oscillations backwards. Once the previous foot has touched the ground the other one starts its movement. We can also notice small modifications on the position and velocity of the ankle joint depending on the different ground slopes. During each stable ground inclination the ankle keeps the oscillatory movement as a periodic signal and changes its strength when the inclination varies. These modifications will alter the body velocity and step's length as shown in Figure 14.

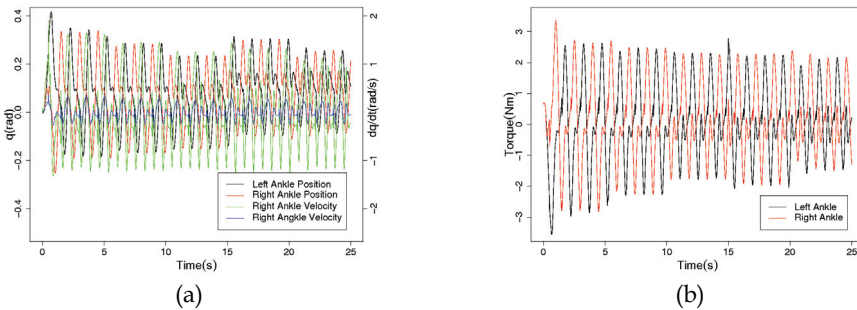


Fig. 13. (a) q , dq/dt and (b) torque values of the variable spring.

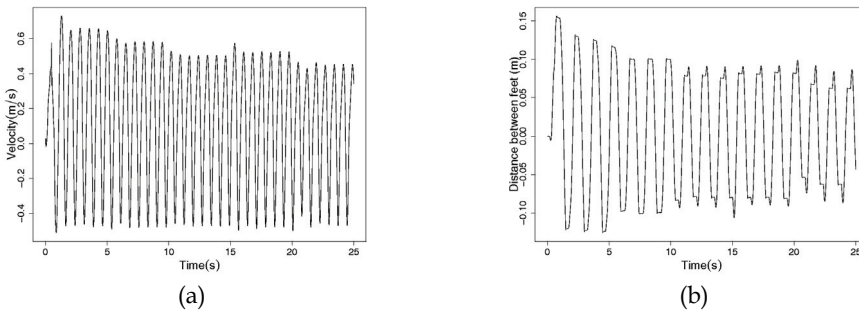


Fig. 14. (a) Body velocity and (b) distance between robot's feet during a trial.

Figure 14a shows that the robot's average speed on a flat terrain is 0.17m/s but it is reduced to 0.08m/s on an 8 degrees ground slope. This variation can also be appreciated in Figure 14b where we observe a starting step size of 0.12m and is reduced to 0.08m at the end of the trial. The maximum distance walked, average speed and average acceleration are shown in Figure 15, related to the instantaneous ground slope.

One of the important characteristics on the robots is its quasi-passiveness which allows the robot to have a reduced consumption as can be appreciated in Figure 16. The instantaneous power consumed by the actuator of the tail has been calculated using equation 15, and the energy by equation 16.

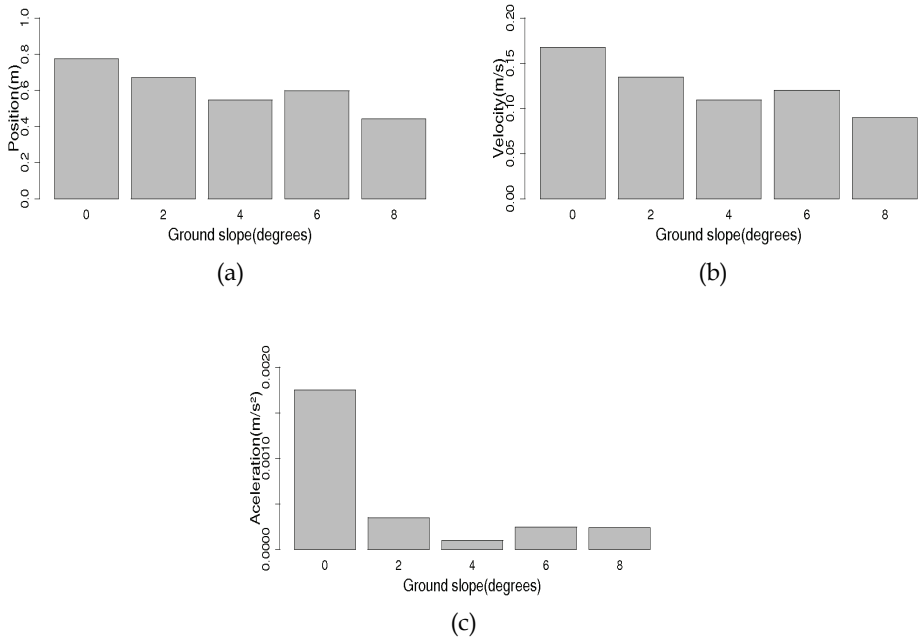


Fig. 15. (a) Distance travelled (b) Average velocity and (c) Average acceleration of the centre of mass of the robot related to each different inclination angle.

$$P(t)=\tau(t)\dot{q}(t) \tag{15}$$

$$E=\int_0^t |\tau(t)\dot{q}(t)| dt \tag{16}$$

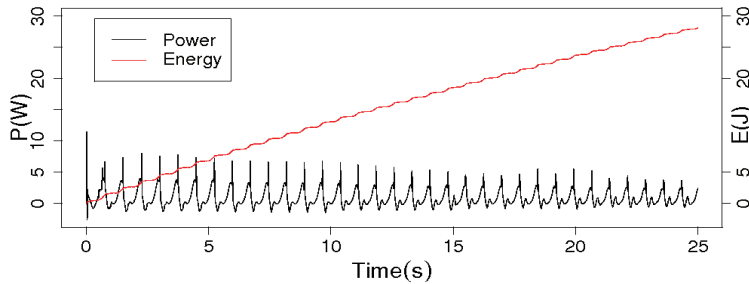


Fig. 16. Tail joint mechanical instantaneous power and energy consumption during a trial. Finishing with the results evaluation we show the limit cycle of the robot during the trial for the different ground inclinations on Figure 17.

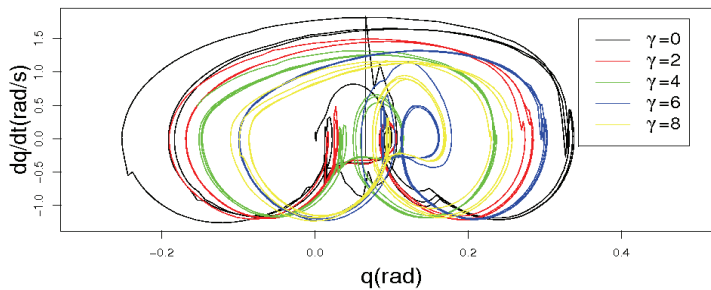
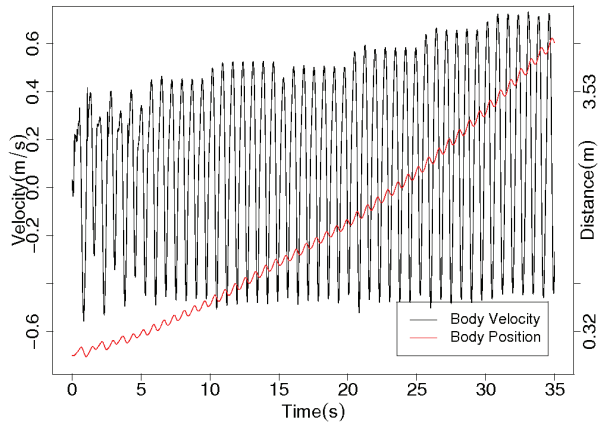


Fig. 17. Limit cycle on different inclinations during a trial.

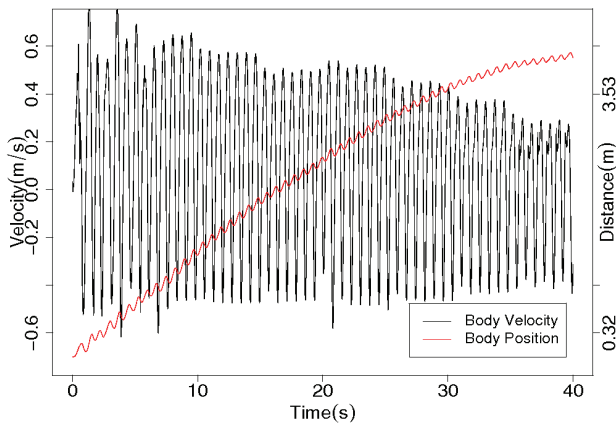
Once the evaluation of the best genotype has been finished, we tried to evaluate it in different situations. Firstly we tried to test the robot on different slopes. Tests show that the robot is able to go up slopes up to 12° and goes down slopes of -2° , but no more.

Later we tried to test initial conditions problems, about seeing if the robot was able to start in different slope conditions. Results show that robots can start in positive slopes up to 10 degrees but no 12 degrees even if it is able to walk over them. The robot can satisfactorily start in -2° slopes.

Finally Figure 18 shows two different trials for these configurations: In (a) robot start over a 10° slope which reduces 2° each 5 seconds. (b) shows a robot starting at a -2° ground slope and finishing in a 12° positive inclination.



a)



b)

Fig. 18. (a) Post-evaluation for a 35 seconds simulation, starting on a 10 degrees slope and going down to -2 degrees. (b) Post-evaluation for a 40 seconds simulation, starting on a -2 degrees slope and going up to 12degrees.

7. Conclusions

It is shown through computer simulations that it is possible for a biped robot with a tail to go up a small slope.

We have followed an evolutionary methodology to tune a continuous-time recurrent neural network. The evolution is made inside an environment that changes suddenly from one slope to another without stopping the robot.

The neural network changes in real-time the parameters of the ankle springs and the oscillation frequency of the tail, without giving an explicit path in the configuration space of the robot. It is also shown that the robot performs well for new conditions of the environment not considered in the evolution phase: for higher slopes and for different initial conditions.

This work almost ends the first necessities investigations to build a new real Zappa robot. New mechanisms for the ankles and modifications on the electronics of the robot are needed. Some studies on the sensors of the robots must be carried out. These studies could offer a better performance on the task and give some perceptions to the robot about its position (inclination) related to the ground on its postural state.

Nevertheless some theoretical studies, both kinematical and dynamical are still needed.

8. References

- Alexander, R. M. (2005). Physiology Walking made simple, *Science Magazine*, Vol. 308, No. 5178 (April, 2005) 58-59, ISSN 0036-8075.
- Asano, F., Luo, Z. & Yamakita, M. Some Extensions of Passive Walking Formula to Active Biped Robots, *Proceeding of the 2004 IEEE International Conference on Robotics and Automation*, pp. 3797-3802, ISSN 1050-4729, New Orleans, LA, April 2004, IEEE Press, Washington D.C.
- Asano, F., Luo, Z. & Yamakita, M. (2005). Biped Gait Generation and Control Based on a Unified Property of Passive Dynamic Walking. *IEEE Transactions on Robotics*, Vol. 21, No. 4, (August, 2005) 754-762, ISSN 1552-3098.
- Beer, R. D. & Gallagher, J. C. (1992). Evolving Dynamic Neural Networks for Adaptive Behaviour. *Adaptive Behavior*, Vol. 1, No. 1 (June, 1992) 91-122, ISSN 1059-7123.
- Beer, R.D. (2006) Parameter Space Structure of Continuous-Time Recurrent Neural Networks. *Neural Computation*, Vol. 18, No. 12, (December, 2006) 3009-3051, ISSN 0899-7667.
- Berenguer, F. & Monasterio-Huelin, F. (2006). Easy design and construction of a biped walking mechanism with low power consumption, *Proceedings of the 9th Int. Conf. on Climbing and Walking Robots*, pp. 96-103, Brussels, Belgium, September 2006.
- Berenguer, F. & Monasterio-Huelin, F. (2007a). Trajectory planning using oscillatory chirp functions applied to bipedal locomotion, *Proceedings of the 4th Int. Conf. on Informatics in Control, Automation and Robotics*, pp. 70-75, ISSN 0738-4602, Angers, May 2007, AAAI, La Canada CA.
- Berenguer, F. & Monasterio-Huelin, F. (2007b). Stability and smoothness improvements for an underactuated biped with a tail, *Proceedings of the 2007 IEEE Int. Symp. on Industrial Electronics*, pp. 2083-2088, ISBN 1-424-40754-0, Vigo, Spain, June 2007, IEEE Press, Piscataway N.J.
- Berenguer, F. & Monasterio-Huelin, F. (2007c). Locomotion of an Underactuated Biped Robot Using a Tail, In: *Bioinspiration and Robotics: Walking and Climbing Robots*, Maki K. Habib (Ed.), 15-38, I-Tech Education and Publishing, ISBN 978-3-902613-15-8, Vienna, Austria.
- Berenguer, F. & Monasterio-Huelin, F. (2008). Zappa, a Quasi-passive Biped Walking Robot with a Tail. Modeling, Behavior and Kinematic Estimation using Accelerometers. *IEEE Transactions on Industrial Electronics*, (September, 2008), (in press).

- Boeing, A. & Bräunl, T. (2004). Evolution of Locomotion Controllers for Legged Robots. *Robotic Welding, Intelligence and Automation (LNCIS)*, No. 299, (2004) 228–240, ISSN 0170-8643.
- Collins, S. H. & Ruina, A. (2005). A Bipedal Walking Robot with Efficient and Human-Like Gait". *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp.1983-1988, ISSN 1050-4729, Barcelona, Spain, April, 2005, IEEE Press, Washington D.C.
- Edwards, W.T. (2007) Effect of joint stiffness on standing stability. *Gait & Posture*, Vol.25, No. 3, (March 2007) 432-439, ISSN 0966-6362.
- Geyer, H. & Seyfarth, A. (2006). Walking and running dynamics explained by compliant legs: consequences, general insights, and future directions. *Journal of Biomechanics*, Vol. 39, Supp. 1, (2006) 361, ISSN 0021-9290.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, ISBN 0-201-15767-5, New York, NY.
- Grossberg, S. (1988) Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, Vol. 1, No. 1, (1998) 7-61, ISSN 4083-6601
- Holland, J.H. (1975). *Adaptation in Natural and Artificial System*, MIT Press, ISBN 0-472-08460-7, Ann Arbor, MI.
- Hopfield, J.J. (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings Of the National Academy of Sciences of the United States of America*, Vol. 81, No. 10, (May 1984) 3088-3092, ISSN 0027-8424.
- Hurmuzlu, Y., Génot, F. & Brogliato, B. (2004). Modeling, stability and control of biped robots—a general framework. *Automatica*, Vol. 40, No. 10, (2004) 1647-1664, ISSN 0005-1098.
- Iida, F. & Pfeifer, R.. Sensing through body dynamics. *Robotics and Autonomous Systems*, Vol. 54, No. 8, (August,2006) 631–640, ISSN 0921-8890.
- Iida, F., Rummel, J. & Seyfarth, A. (2007). Bipedal Walking and Running with Compliant Legs, *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, pp. 3970-3975, ISBN: 2-130-31308, Rome, Italy, April 2007, IEEE Press, Piscataway, NJ.
- Iida, F., Minekawa, Y., Rummel, J. & Seyfarth, A. (2008). Toward a Human-like Biped Robot with Compliant Legs. *Robotics and automation systems*. (2008), (in press).
- Kluzik, J., Horak, F.B. & Peterka, R.J. (2007a). Postural After-effects of Stepping on an Inclined Surface. *Neuroscience Letters*, Vol. 413, No. 2, (February 2007), 93-98, ISSN 0304-3940.
- Kluzik, J., Peterka, R.J & Horak, F.B..(2007b) Adaptation of postural orientation to changes in surface inclination. *Experimental Brain Research*, Vol. 178, No. 1, (March 2007) 1-17, ISSN 0014-4819.
- Kuo, A. D. (2007). The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective. *Human Movement Science*, Vol. 26, No. 4, (2007) 617-656, ISSN 0167-9457.
- Löffler, K., Gienger, M., Pfeiffer F. & Ulbrich, H. (2004). Sensors and Control Concept of a Biped Robot. *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 5 (October 2004) 972-980, ISSN 0278-0046.
- Magdalena, L. & Monasterio, F. (1995). Evolutionary-based learning applied to fuzzy controllers. *Proceedings of 1995 IEEE International Conference on Fuzzy Systems: the*

- international joint conference of the Fourth IEEE International Conference on Fuzzy Systems and the Second International Fuzzy Engineering Symposium*, pp. 1111 - 1118, ISBN: 0-780-32462-5, Yokohama, Japan, (March, 1995), IEEE Press, Piscataway, NJ.
- Matsushita, K., Yokoi H. & Arai, T. (2006). Pseudo-passive dynamic walkers designed by coupled evolution of the controller and morphology. *Robotics and Autonomous Systems*, Vol. 54, No. 8 (2006) 674-685, ISSN 0921-8890
- Maurer, C. & Peterka, R.J. (2005). A New Interpretation of Spontaneous Sway Measures Based on a Simple Model of Human Postural Control. *Journal of Neurophysiology*, Vol. 93, No. 1, (January, 2005) 189-200, ISSN 0022-3077.
- McCulloch, W.S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Vol. 5, No. 4, (December, 1943), 115-133, ISSN 1522-9602
- McGeer T. (1990). Passive Dynamic Walking. *The International Journal of Robotics*, Vol. 9, No. 2, (1990) 62-82, ISSN 0278-3649.
- McMahon, T. A. (1984). *Muscles, reflexes, and locomotion*, Princeton University Press, ISBN 0-691-08322-3, Princeton, NJ.
- Nolfi, S. & Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of self-Organizing Machines*, MIT Press, ISBN 0-262-14070-5, Cambridge, MA
- Ohashi, E., Aiko, T., Tsuji, T., Nishi, H. & Ohnishi, K. (2007). Collision Avoidance Method of Humanoid Robot with Arm Force. *IEEE Transactions on industrial electronics*, Vol. 54, No. 3, (June, 2007) 1632-1641, ISSN 0278-0046.
- Pfeifer, R. and Bongard, J. (2007) *How the body shapes the way we think. A new view of Intelligence*, MIT Press, ISBN 0-262-16239-5, London England.
- Prentice S.D., Hasler, E.N., Groves, J.J. & Frank, J.S. (2004). Locomotor adaptations for changes in the slope of the walking surface. *Gait and Posture*, Vol. 20, No. 3, (December, 2004) 255-265, ISSN 0966-6362.
- Rumelhart, D.E. & McClelland, J.L. (1986). *Parallel Distributed Processing. Vol1 : Foundations.*, MIT Press, ISBN 0-262-18120-7, Cambridge, MA
- Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B. & Lefeber, D. (2007). MACCEPA, the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, Vol. 55, No. 3, (2007) 761-768, ISSN 0921-8890.
- Vidyasagar, M. (1993). *Nonlinear Systems analysis*, Prentice-Hall, ISBN: 0-13-623513-1, Englewood Cliffs, NJ.