

## Gait Synthesis in Legged Robot Locomotion using a CPG-Based Model

J. Cappelletto, P. Estévez, J. C. Grieco, W. Medina-Meléndez  
and G. Fernández-López  
*Mechatronics Research Group, Universidad Simón Bolívar*  
Venezuela

### 1. Introduction

Biology has always been a source of inspiration and ideas for the robotics community. Legged locomotion problem is not an exception, and many experiences have taken ideas from animals, both for morphological and behavioral issues. The first ideas for gait generation came from animal observation, but they were mainly focused on mimicking legs movements. It was not until the nineties that the first relevant works appeared trying to identify the principles behind the generation of those movements in animals. The proposed models were based on neurophysiologic principles, and most of them tried to include characteristics of animal locomotion by the addition of neural networks, dynamic oscillators, or using a set of “movement rules”. Although many models have been suggested, most of them share some common aspects:

1. Motion signals generation and processing are very slow and highly distributed processes.
2. The brain tends to perform high level feed-forward movement control and prediction.
3. The locomotion system has local feedback, from pressure sensors, force sensors, intramuscular sensors, etc.

In some processes these characteristics are obvious, like in the heart beating or breathing. In these processes there is no need for the intervention of a complex processing unit like the brain, since most of the coordinated oscillatory behavior of the muscles is carried out locally and distributed. The oscillatory nature of locomotion patterns has attracted studies about the existence of a similar structure in charge of this problem. The biologic and electrochemical bases of the system in animals are fairly well explained in the works on neural networks by Hodgkin-Huxley (Hodgkin, 1952). Another important characteristic of animal systems is that biological neural networks can perform timing tasks through oscillatory networks, and also can modulate neuromuscular excitatory signals, thus giving the ability to *shape* neural network output. Compared to robot legged locomotion, it is possible to identify some common issues like inter-leg coordination that can be thought as a timing system, and the intra-leg actuators coordination that can be addressed as a spatial system.

In legged animal locomotion the periodical excitation of the flexor and extensor muscles is needed in order to produce effective walking movements. To model this process, two

Source: Bioinspiration and Robotics: Walking and Climbing Robots, Book edited by: Maki K. Habib  
ISBN 978-3-902613-15-8, pp. 544, I-Tech, Vienna, Austria, EU, September 2007

different approaches have been developed. The first one, usually called “reflex chain”, lies on the idea of a complex network of joint sensors connected directly to the actuators, so that the muscles were able to detect the right moment to act in order to produce the desired motion. Based on this, it is possible to model many movement processes by using local feedback control networks. The second approach suggests the idea of an oscillating system located at the low level nervous system, that generates timing signals needed to activate the repetitive actions (Marder, 2001; Marder, 2005), which is called *central pattern generator*.

Nowadays, it is widely accepted that motion generation and control processes are performed in the spinal chord by the Central Pattern Generator (CPG). Combining high level brain signals with low level sensory feedback, this system is able to coordinate the neuromuscular excitation in order to achieve desired leg movements. The CPG can be described as a circuit able to produce rhythmic motor patterns in the absence of any external stimuli. This means that a CPG is present when the existence of a rhythmic behavior does not depend of the peripheral or central nervous system signals, although those signals can alter the motor patterns generated by the CPG. There are several works that study the effect of sensory feedback in the modulating action (Conradt, 2003; Kimura, 2002; Kuo, 2002; Ijspeert, 1999).

Some authors have proposed to model the CPG using continuous time recurrent neural networks (CTRNN) (Gallagher, 1999; Ghallager, 1999); and it has been proved to be a good choice thanks to the CTRNN ability to model dynamic systems and generating rhythmic responses. Other works are based on neural networks built on *leaky integrators* which using local feedback of joint angles were able to synthesize different walking modes for simple legged robot models (Billard, 2000). There are some models that generate a CPG through coupled oscillators described by a set of differential equations as the Amplitude Controlled Phase Oscillator (ACPO) concept developed in the BIRG group of EPFL (Buchli, 2004). The modeling of the walking system with CPG has also been employed in non-legged robots, like snake-like robots (Conradt, 2003). It was possible to control a robot with a high number of DOF actuated through servomotors, whose angle references were generated using a distributed CPG synthesized with coupled oscillators. It also must be mentioned the work developed by R. M. Ghigliazza (Ghigliazza, 2004) based on biomechanical locomotion model on insects like cockroaches. They were able to control the support factor and frequency on leg movement by using a reduced model of recurrent networks implemented through coupled oscillators. Many solutions have been proposed for robots with simple leg models, but several key functionalities remain unsolved in a clear way, like coordination of articulations movements for legged robots with medium-high complexity in the kinematical structure and marginally stable platforms like 3 DOF quadrupeds and hexapods. It is also necessary to provide models where temporal reference generation and spatial coordination can be controlled in a separate way mimicking the biological model, and thus giving the ability to apply specific control actions that does not require major modifications on the system architecture when migrating through different legged platforms.

This chapter describes a technique for gait synthesis on legged robots by using the Central Pattern Generator (CPG) concept. The work here described bases on separating the walking problem in a temporal coordination and a spatial representations problem. By the addition of a nonlinear space transformation subsystem, it is possible to convert reference vector from the temporal coordination subsystem, into spatial references in the legs workspace, with the capability to manage all the spatial corrections (obstacles, stability problems, direct or inverse kinematics). Sensor feedback can be employed in both subsystems in order to

control actions. The chapter will be organized as follows: an introduction of the state-of-the-art review in robot locomotion models based on CPG; section two will describe the basic system architecture, following the principle of separation into temporal coordination and spatial reference generation subsystems; section 3 describes a first approach for the CPG model using simple time references and FFNN, including experimental setups and results. Sections 4 and 5 introduce two improved models to solve drawbacks observed in the previous model; they are based on coupled oscillators and employing FFNN and parametric trajectory description for the spatial subsystem. Conclusion and future works are shown in section 6, and finally references will be at section 7.

### 1.1 Review on Legged Robot Locomotion using Central Pattern Generator (CPG).

As described in the previous section, CPG models can be separated on those based on a recurrent neural networks architecture, including *leaky integrators*, which are models derived of simple Amari-Hopfield oscillator model and CTRNN, and those that rely in coupled differential equations, like ACPO.

The model proposed by Chiel, Beer, and Gallagher (Gallagher, 1999) is based con CTRNN, built on neuron nodes which transfer function is given by Equation 1.1.

$$\tau_i \dot{y}_i = -y_i + \sum_{j=1}^M \omega_{ji} \sigma(y_j + \theta_j) + I_i, \quad i = 1, \dots, M \quad (1.1)$$

In the Eq. 1.1 the variable  $y$  is the output vector state for the  $i^{\text{th}}$  neuron. The time constant associated with the cellular membrane permittivity is represented by  $\tau$ , while the synaptic connections weights are represented by  $\omega_j$ . The variable  $\theta$  is the node bias point. The kernel of the transfer function of each neuron is the standard logistic sigmoid, given by the following Equation:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.2)$$

There are several important characteristics in this particular model of neural networks. The Eq. 1.1 can be interpreted as describing the temporal evolution for the state variable  $y$  where it can converge to a single attractor point, multiple attractor point, or a closed attractor cycle. The most used interconnection architecture for this kind of systems is fully bidirectional interconnected networks. The number of neurons  $N$  in each network used to be between 3 and 5. Although bigger networks could be implemented to achieve richer dynamics, their analysis complexities make them impractical.

The use of Genetic Algorithms (GAs) for the synthesis of parameters for the CPG based on CTRNN was mentioned in *Gallagher et al.* work. As fitness function, it was used the robot body speed obtained by evaluating the resulting network as a torque reference generator for the leg actuators. The kinematical model employed by Gallagher (Gallagher, 1999) consists in a 2 DOF leg, with two neurons controlling swing (FS and BS), and other neuron controlling the vertical positioning of the foot (FT). Figure 1.1 shows this CPG architecture based on CTRNN.

This model provides a direct link between the neural oscillator and the actuator references, and can be modified to other leg kinematics. Satisfactory results have been reported using

this approach in relatively simple body models. However, the parameter synthesis process cannot assure convergence to a desired leg movement. Also, the interconnection between legs remains unsolved in a direct way because the variation of the dynamic for a leg cannot be predicted when it is connected with another neural module. These aspects are important drawbacks for employing this model of recurrent neural network for modeling the locomotion system in legged robots.

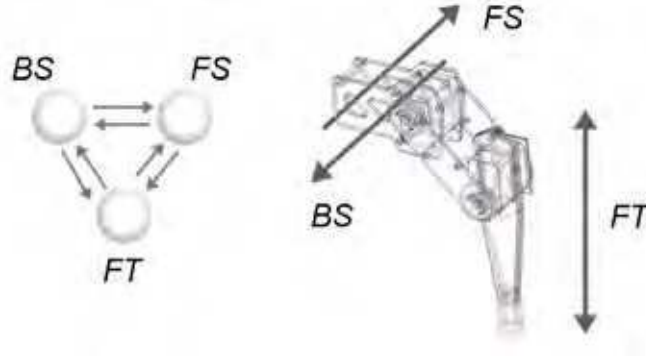


Figure 1. CTRNN based CPG architecture

The model based in coupled oscillators as described by J. Buchli (Buchli, 2004) provides another approach to control, in a direct way, the phase relations between robot legs. The main concept of this model is the ACPO, which can be seen as a phase locked oscillator. It is based on a network of fully interconnected nodes, with interconnection weights and a rotation matrix that control the phase relations between each oscillatory node. Each node is represented by a 2-dimensional state variable  $q = [x \ y]$ . The state transition function for a perturbed oscillator is given by Eq. 1.3.

$$\dot{q} = F(q) + p \quad (1.3)$$

Equation 1.3 shows how the derivative of the vector state depends on the natural unperturbed response of the system through the state transition function  $F$ , and also depends on the external system perturbation  $p$ . For the unperturbed state ( $p = 0$ ), the natural system response must be oscillatory. Working with a two dimensional system, a space transformation into radius ( $r$ ) and phase ( $\theta$ ) can be applied. Given this, the perturbation can be separated into a radial ( $p_r$ ) and a tangential ( $p_\theta$ ) component. For robot locomotion it is important the phase relation between oscillators, and due to this reason the  $p_\theta$  component is more relevant in the ACPO analysis. In (Buchli, 2004) it is described the phase locking between two oscillators, and how to predict the temporal evolution of  $\theta_d$  for a given pair of oscillators and its coupling relation. The oscillatory limit cycle is described by Eq. 1.4:

$$\dot{q} = \left[ g \left( \frac{r_0}{\sqrt{x^2 + y^2}} - 1 \right) x - y\omega, g \left( \frac{r_0}{\sqrt{x^2 + y^2}} - 1 \right) y - x\omega \right]^T \quad (1.4)$$

For this equation the state vector  $q$  converges to a circle of radius  $r_o$ , with a natural frequency of  $\omega$ . The variable  $g$  denotes the convergence gain to the limit cycle. Being  $F_{ACPO}(q)$  the limit cycle described by Eq. 1.4, and modeling the interconnection between nodes like perturbations, the full CPG for a quadruped robot can be described as:

$$\begin{aligned}\dot{q}_1 &= F_{ACPO}(q_1) + p_c(q_2) + p_c(q_3) + p_c(q_4) \\ \dot{q}_2 &= F_{ACPO}(q_2) + p_c(q_3) + p_c(q_4) + p_c(q_1) \\ \dot{q}_3 &= F_{ACPO}(q_3) + p_c(q_4) + p_c(q_1) + p_c(q_2) \\ \dot{q}_4 &= F_{ACPO}(q_4) + p_c(q_1) + p_c(q_2) + p_c(q_3)\end{aligned}\quad (1.5)$$

where  $p_c(q_i)$  is the perturbation vector produced by the coupling among each node, and it is obtained by rotating each  $q_i$  vector according to the desired phase relations between oscillators.

It can be observed that this model provides a direct control of oscillatory frequency, and phase relations between legs, however the system outputs are mainly sinusoidal signals that cannot drive joints actuators directly. It is necessary the inclusion of a nonlinear space transformation that converts these phase locked temporal references into useful motor references according to the leg kinematical structure.

## 2. Proposed Legged Robot Locomotion Models Based on CPG

This section describes different implementations for a CPG based model, with a system architecture that follows the philosophy of separating the walking problem into two different, but not unrelated, spatial and temporal subsystems. This approach is derived from observations of biological systems, where locomotion is associated with rhythmic behavior of neuromuscular activity giving the timing reference, and local modulation is done as consequence of the sensory feedback. Such situation has its analogy with robot locomotion where it is required a control of phase relations between legs, and specific spatial control to cope with leg kinematics and any perturbation due to interaction with the environment. It is important to point out that the spatiotemporal separation allows the implementation of better control schemes because issues like stability, gait modes, weight distribution and others, and can be addressed by applying control actions well developed for standard approaches in robot locomotion.

The design process of the locomotion system starts by pointing the desired characteristics for the gait synthesis model, which can be identified as:

1. **Reference trajectories for legs:** The movement of each leg must describe a continuous closed shape. Such trajectory consists on a transfer phase and a support phase, which provides the effective propulsion for the robot platform. The spatial references are associated with the leg kinematics and are desirable to be in the actuators space, as joint angles, for example. By controlling this stage it is possible to deal with uneven and irregular terrains.
2. **Inter-leg movements coordination:** The specific robot gait is given in a direct way by the phase relation between the leg movements. It is desirable that the model provides a way to make soft transitions between different walking modes.

### 3. Proposed Model for Gait Generation: Simple Temporal Reference and FFNN

This first model is centered in the use of rhythmic signals for the temporal coordination, and a feed-forward neural network (FFNN) for the spatial reference subsystem. The FFNN is employed to perform a nonlinear space transformation from the simple low dimensional time signal to higher dimensional DOF references; also, it is possible to model the neural modulation phenomenon observed in biological systems, by making soft transitions between different referential trajectories for the tip of the leg. More details on this approach are given by Cappelletto et. al. (Cappelletto, 2006) where recent works on gait synthesis using FFNN are described.

#### 3.1 System Architecture and Experimental Setup

For the temporal reference subsystem, three different timing signals were employed: a cyclical ramp, a two dimensional CTRNN output state vector, and a 2D circular vector, as shown in Figure 2.

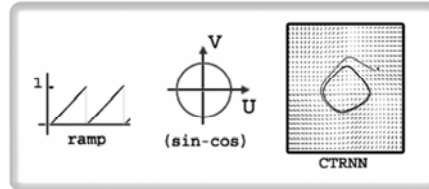


Figure 2. Temporal references

For the 2D circular vector it was employed a simple sine-cosine pair described by Equation 3.1. As CTRNN, a two neuron network was used with the same node equation as Eq. 1.1 and 1.2.

$$\begin{cases} U = A \cdot \sin(\omega_s t + \phi_s) \\ V = B \cdot \cos(\omega_c t + \phi_c) \end{cases} \quad (3.1)$$

The parameters employed for the 2D vector (UV) are:  $A = B = 1$ ,  $\omega_s = \omega_c = \omega$  and  $\theta_s = \theta_c = \theta$ . With these values, it is obtained a perfect circle with unit radius, centered at the origin, and with a constant rotation of  $\theta$ . A GA is used to synthesize the CTRNN parameters. Because the CTRNN is employed as a pacemaker, it is enough condition that the neural network output oscillates at a given frequency  $\omega$ .

For the nonlinear space transformation it is employed a two layer feed-forward neural network with standard sigmoid transfer function, trained using backpropagation. The input vector has dimension  $M$  and the output vector has dimension  $N = 3$  corresponding to a 3 DOF leg model. For the hidden layer of the FFNN it is employed  $K = 18$  neurons. At the output level, a linear transformation is required from bounded neuron outputs to joint servomotor angle references. The physical robot is a 3DOF per leg small quadruped built by Lynxmotion®, with reptile-like leg posture as can be observed in Figure 3. The main body dimensions are  $L = 240mm \times W = 190mm$ . Each joint is directly actuated with servomotors,

and the leg links dimensions, from shoulder to the tip of the leg, are  $L1 = 33mm$ ,  $L2 = 70mm$  and  $L3 = 113mm$  respectively.

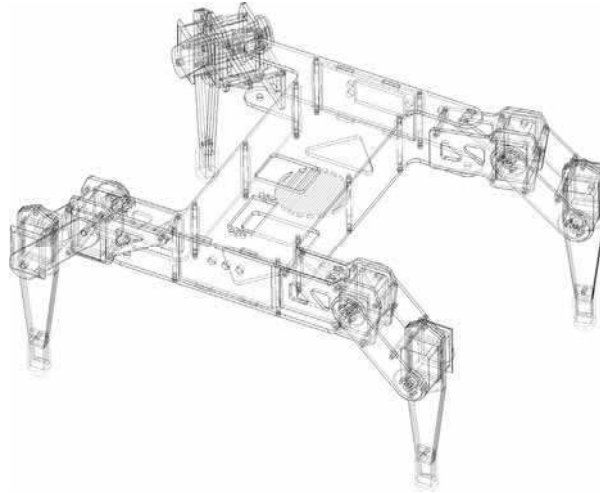


Figure 3. Quadruped robot model

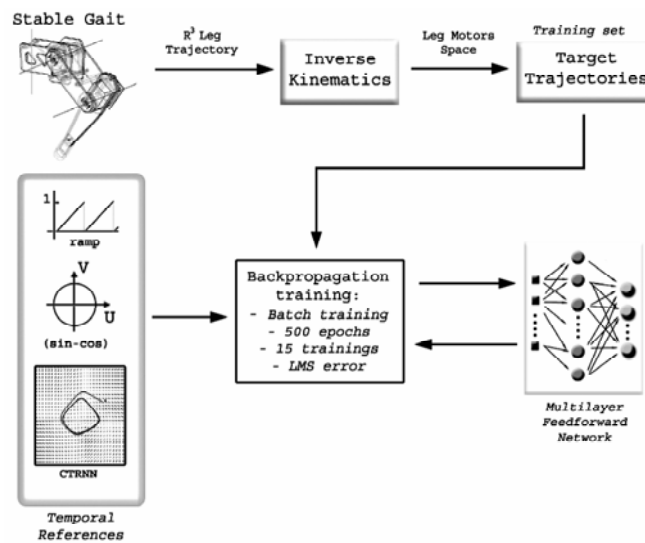


Figure 4. FFNN Training scheme

For the training process, statically stable leg trajectories are converted into desired joints angles references via inverse kinematics. Those resulting waveforms are employed as target outputs of the FFNN. An additional input *mode* was included to the neural network in order to choose the desired shape of the spatial reference. Three different shapes are evaluated: triangle, rectangular, and rectangle with rounded corners. The complete training scheme is shown in Figure 4. Standard batch backpropagation was employed as training method with

1000 points per batch, using NNTOOLS in Matlab ®. The Least Mean Square (LMS) error metric was employed to compute the neural network output error. The backpropagation process was applied during 500 epochs or until a LMS error lower than 2% was reached. Other training methods can be applied to obtain higher convergence rates and best computational performance. This technique is employed to obtain an acceptable solution for nonlinear transformation of system references.

### 3.2 Experimental Results

In Figure 5 it is shown the temporal evolution FFNN training process with the three different temporal references, for a fixed desired output shape (rectangle). It can be observed that training process eventually converges to a low error solution. The best convergence speed is obtained for the UV oscillatory vector, with the CTRNN having an acceptable performance. However for the *ramp* signal, the error decrease rate is lower than other solutions, due to the closed cycle nature of 2D dimensional temporal references employed, that are similar to ones exhibited by leg movement.

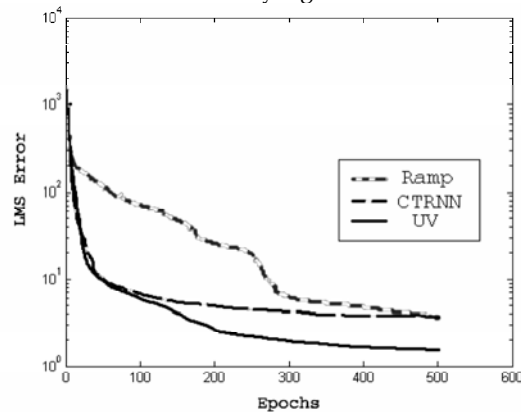


Figure 5. Temporal evolution of FFNN training

The other important experiment is to test the ability of the FFNN to perform the modulation of the spatial leg reference. Using an UV input as temporal reference, the network had the input *mode* ranging from 1 for the rectangular shape to 3 for the triangular one. After applying backpropagation under similar conditions that those employed for the previous experiment, it was observed that correctly trained networks can make soft transitions between the desired leg references. However, it was also obtained that overfitting can be the main problem for this approach because the FFNN ability to modulate is degraded significantly, as can be observed in Figure 6, where a solution with lower LMS error (right image), has an observable degradation in modulation performance.

Using a FFNN without overfitting, with UV temporal reference and *mode* input value for a rounded rectangle shape, it was possible to generate a functional walking pattern. The phase relations between UV oscillators were fixed to those required for *crawl* gait. The motor pattern was tested on the real robot platform and exhibited a marginally stable behavior. The main drawback with this first model is the lack of any kind of interconnection between leg oscillators, as observed in biological systems; so the generation of different gait modes cannot be obtained without further modifications of this model.



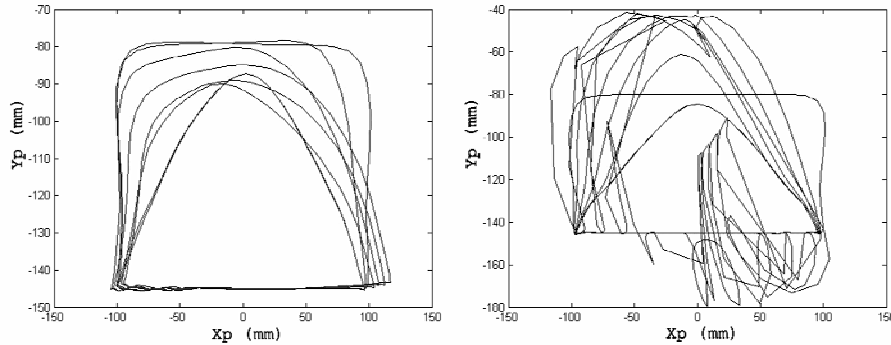


Figure 6. Shape modulation and overfitting

#### 4. Proposed Model for Gait Generation: ACPO and FFNN

In order to overcome the inability to generate different gait modes of the previous model, it was included a set of ACPO acting as pacemakers. The FFNN is employed for the space transformation, thus solving the drawback of coupled oscillators to generate valid spatial references for the leg. In this new approach, it is maintained the main system architecture following the idea of spatiotemporal separation. By the exclusion of a *mode* input is possible to evaluate the performance of this new model without caring about the overfitting problem, which could be addressed using better training processes. In standard geometric models of legged robot locomotion there is a parameter that controls the time of effective support given by each leg, and it is called *support factor* ( $\beta$ ). In this model,  $\beta$  is included between ACPO outputs and FFNN inputs, so it can be controlled without any additional network training or architecture modification. Further details in this model can be obtained in (Cappelletto 2006).

##### 4.1 System Architecture and Experimental Setup

The main system distribution is similar to that employed for the previous model. At the temporal reference there is a set of four coupled oscillators that conforms the ACPO. Each node state vector is passed through a companding curve that modifies its phase according to the support factor  $\beta$ , thus providing a direct control over this parameter. The vector modulus is kept unmodified. The resulting vector is feed to the FFNN that performs a nonlinear space transformation into direct joint angle references. The output layer of the neural network is built on linear neurons instead the sigmoid transfer function utilized for the first model. By this way we avoid the use of any extra stage for linear conversion into valid angle references. The complete system architecture can be appreciated in Figure 7.

It can be observed that the parameters that describe a specific gait mode, are decomposed on those affecting spatial system, like the ones associated to the desired leg trajectory, and those describing the gait mode and speed. The last ones are fed to the temporal subsystem, and are modeled through the phase coupling matrix of the ACPO, attractor cycle angular speed and support factor. For this specific model, it is not used soft transition functions for any change on phase relations due to gait mode switch, as originally cited by J. Buchli (Buchli, 2004). Due to differential nature of the description of coupled oscillators, there will be always a continuous trajectory for the phase component of  $q$  vector, even for abrupt phase

reference changes. As robot model, it was employed the same Lynxmotion quadruped robot described in the previous section, and a companding curve was developed to perform the support factor control. The equation for the phase transformation is:

$$c(x, \beta) = \begin{cases} \frac{x}{2\beta} & x \leq \beta \\ \frac{x}{2(1-\beta)} + \frac{1}{2} \left( 1 - \frac{\beta}{1-\beta} \right) & x > \beta \end{cases} \quad (4.1)$$

In Equation 4.1, the  $x$  input denotes the original phase of each ACPO node, which is converted using two rectilinear segments, with slopes controlled with the support factor  $\beta$ . The resulting transfer function for this companding function is shown in Figure 7.

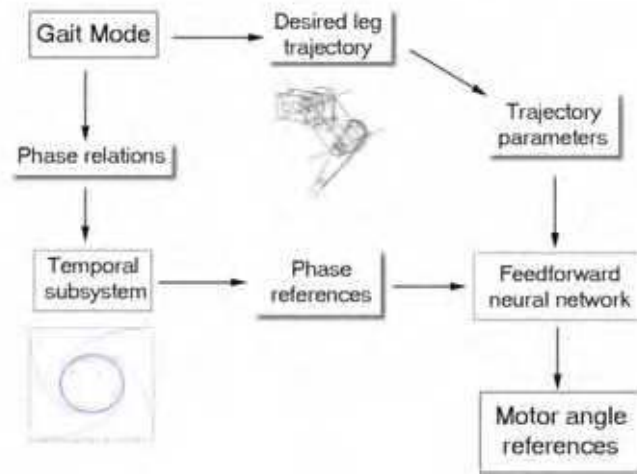


Figure 6. System Architecture ( ACPO + FFNN )

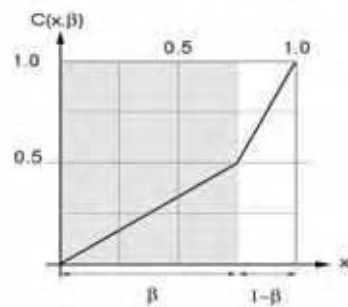


Figure 7. Phase companding curve

At the FFNN level, there is a two layer network with sigmoid neurons in the hidden layer, and linear neurons for the output layer. The training process is point-by-point

backpropagation, no *momentum* added. The target vector consists in 100 points randomly distributed over the references leg trajectory converted into actuators space. The total number of iterations goes from 500,000 to 2.5 millions. For this model, the overfitting phenomenon does not represent a problem for gait generation because there is no need of soft shape transition between different spatial references.

For platform stability improvement it is added a *displacement factor (DF)* that represents an offset in leg tip position over the plane of locomotion. By this way is possible to improve static stability margin, given by the vertical projection of the center of gravity of the body, onto the support surface ( McGhee, 1968 ). This addition shows the flexibility of the model to include well known control actions in walking models based on geometric descriptions.

#### 4.2 Experimental Results

In order to verify the model ability to generate valid walking patterns, is necessary to test the leg references generation using neural networks. The important parameters in the FFNN are the number of hidden units, and the number of training iterations. Table 1 shows five different conditions for NN training. The number of hidden neurons  $K$  varies from 6 to 25, and the number of iterations are 2 millions or 8 millions, for the last network.

	K ( Hidden Neurons)	N° of iterations
NN1	6	2 millions
NN2	8	2 millions
NN3	18	2 millions
NN4	25	2 millions
NN5	25	8 millions

Table 1. Trained FFNN

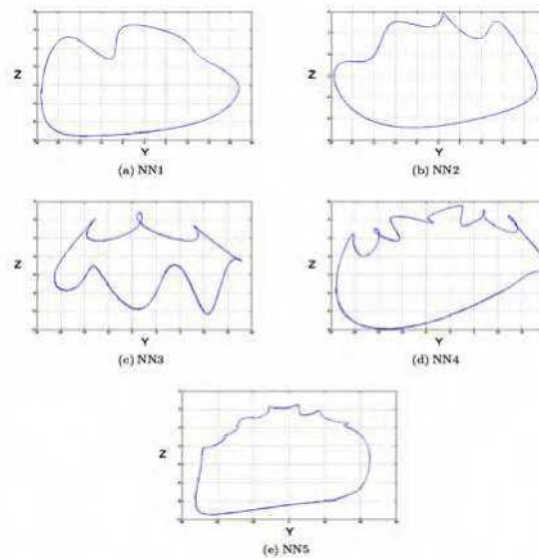


Figure 8. Output trajectories for trained FFNN

Testing each network, by feeding them with the output of a single ACPO node, it was obtained that resulting waveforms, once it was applied the direct kinematics to convert angle references into space references (see Figure 8). The figures are in Z-Y plane which is parallel to leg movement, and perpendicular to support plane (X-Y).

In all trained NN, the output waveform contained oscillatory components, with a frequency that increases with the number or hidden neurons, and this is due to the relation between  $K$  and the number of coefficients presents for the waveform approximation task. Those oscillations also decrease with the number of iterations, because the LMS error is reduced. However, the presence of this behavior is undesired because it can degrade walking performance by introducing mechanical vibrations, and reduces the platform stability.

For the ACPO, there is another issue that can degrade model performance. When the gait mode is changed, the phase between output state vectors maintains the desired relations; however there are noticeable changes in vector modulus as can be observed in Figure 9. This can be solved by applying a normalization stage before feeding the FFNN with the ACPO output.

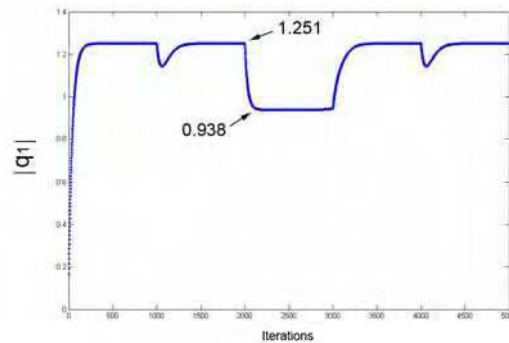


Figure 9. ACPO vector magnitude through time

The resulting CPG can control the real quadruped platform, and describes a marginally stable gait. The addition of the displacement factor  $DF$  makes possible to improve the stability margin and can overcome small irregularities in weight distribution in the platform.



Figure 10. Vertical accelerations per leg

Figure 10 shows the vertical accelerations measured on each leg shoulder, and verify the presence of noticeable oscillations introduced by the neural network and amplitude variations of ACPO nodes for gait changes.

## 5. Proposed Model for Gait Generation: ACPO and Parametric Trajectories

This model solves oscillation and instability problems by replacing the feedforward neural network with a parametric description of the leg trajectory. The reference signal for the spatial subsystem is the ACPO nodes phase, instead of  $x$ - $y$  components of such two dimensional vectors. By the addition of normal contact force feedback it can be improved stability margin for different gaits, for quadruped and hexapod platform with 3 DOF. The main system architecture remains unchanged, except the spatial subsystem where the FFNN is no longer used as it was pointed out previously. (Cappelletto, 2007; Cappelletto et. al., 2007).

### 5.1 System Architecture and Experimental Setup

As the two previous models, this approach keeps the separation between spatial and temporal subsystems. The compensating curve for support factor control is kept, and it is included a force feedback loop to improve stability margin. This structure can be appreciated in Figure 11. It can be observed the addition of a Pressure Center Reference Generator (PCRG) that is fed with ACPO phase outputs and desired motor angles. The PCRG generates the reference for the force control loop that modifies the DF in the final legs trajectories. This loop control enhances platform stability by increasing the distance between measured center of gravity of the robot, and sides of the support polygon thus augmenting stability according to McGhee criterion (McGhee, 1968).

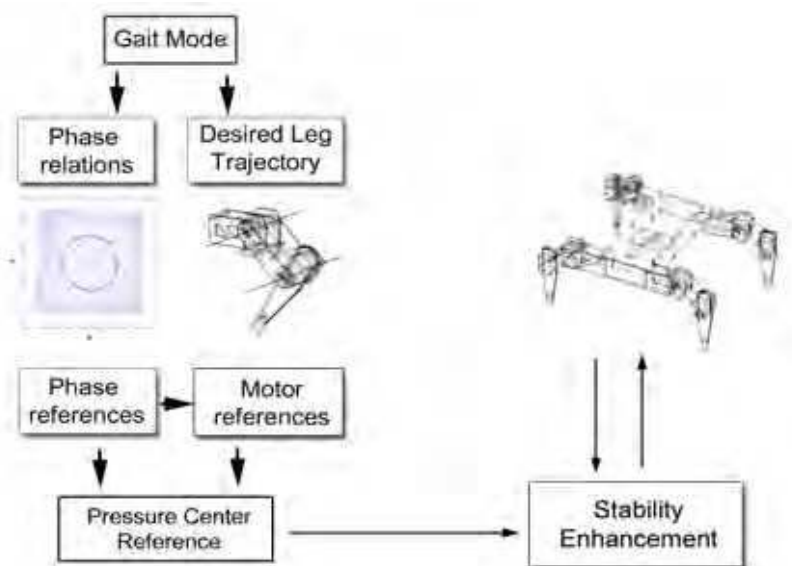


Figure 11. System architecture. CPG model with force feedback

By employing only the ACPO vectors phase, instead the  $x$ - $y$  components, the effect of amplitude variations due to gait changes is neglected, thus improving system performance. In order to control an hexapod platform, the original ACPO nodes were extended to deal with the six legs. The interconnection schemes required for quadruped and for hexapod platforms are shown in Figure 12; in the case of the quadruped is possible to synthesize the standard gaits like crawl, gallop and run, and for the hexapod is possible to generate directly undulatory gaits. All dynamic simulations were done using Webots ® tool. As hexapod model it was employed a body with dimensions of 335 x 150 mm. The hexapod legs are exactly the same modeled for the real and simulated quadruped robot.

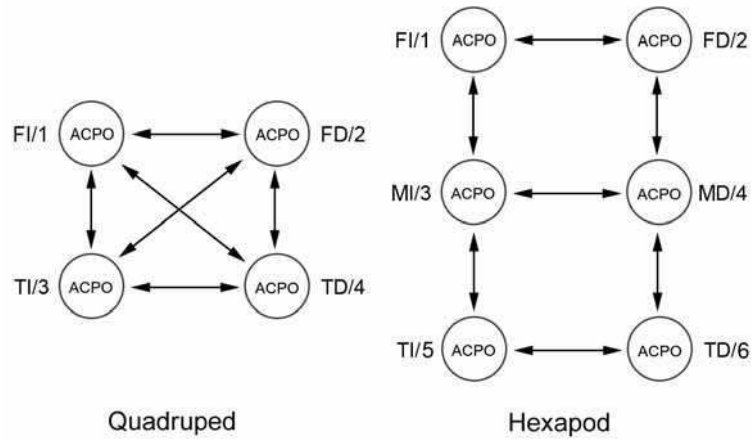


Figure 12. Interconnections schemes for quadruped and hexapod

In the specific case of the hexapod, the connections for opposite legs (1-2, 3-4, 5-6) have a fixed phase of 180 degrees, and connections for adjacent legs (1-3, 3-5, 2-4, 4-6) have a phase that depends on support factor  $\beta$ .

For the force control loop, there is a PCRG that can be implemented with different geometric or force based schemes. In this specific implementation, there are three different kind of PCRG. The first one, named Balanced Forces Point (BFP) calculates an average of all supporting leg tips positions using their referential forces as weights (Equation. 5.1). The legs on transfer phase are naturally ruled out due to their null force reference, and the slopes in the force references allows soft transitions between changes of the BFP. The BFP is always located inside the convex hull of the support polygon, and gives a balanced distribution of effort among the legs.

$$X_{\text{BFP}} = \frac{\sum_{i=1}^N X_i \cdot P_i}{\sum_{i=1}^N P_i} ; Y_{\text{BFP}} = \frac{\sum_{i=1}^N Y_i \cdot P_i}{\sum_{i=1}^N P_i} \quad (5.1)$$

It is easy to obtain support legs distributions yielding to a location of the BFP with suboptimal Static Stability Margin (SSM). However, experiments show that for the kind of

support distributions usually found in legged platforms and for small number of legs, the BFP shows and acceptable performance.

In the second algorithm the desired convex support polygon is identified by using the referential leg forces, and calculates its Area Centroid (AC). This point will be always contained into the support polygon due to its natural convexity (Equation 5.2). This solution provides a balanced distribution of the support polygon because the AC generates a reference located at a balanced distance of the polygon borders.

$$\vec{r}_{CA} = \frac{\int_{\text{polygon}} \vec{r} \cdot da}{\int_{\text{polygon}} da} \quad (5.2)$$

The third algorithm tries to overcome with computational complexities present in the AC method, while keeping the most important variable that are distance to support polygon borders. The employed equation (5.3) is a slight modification of the previous one, and is computed using the polygon contour instead the whole area.

$$\vec{r}_{CC} = \frac{\oint_{\text{contour}} \vec{r} \cdot d\ell}{\oint_{\text{contour}} d\ell} \quad (5.3)$$

In order to calculate the real COG of the robot, normal force sensors (Flexiforce™) are placed at the tip of each robot leg. Using this sensor information and joints angle, it is possible to compute the COG using equation 5.1. Based on measured position of COG X and Y coordinates, and using the desired coordinates obtained from the PCRG, are generated two error signals that are connected to the control system shown in Figure 13. The controller is a Proportional-Integrative one.

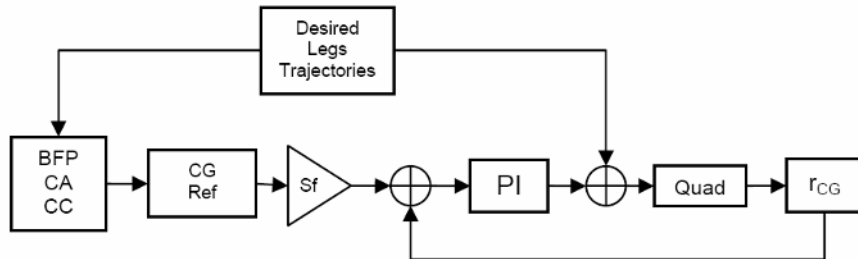


Figure 13. Force based control scheme

## 5.2 Experimental Results

Using the model previously described it is possible to synthesize several gait modes for both simulated and real quadrupeds, and for a simulated hexapod. The performance of the model for the SSM values using different PCRG as control references can be evaluated in Table 2. It is also included the results for measured SSM when control loop is disabled.

Test conditions	Average SSM (mm)			
	BFP	AC	CC	No Control
Quadruped $\beta = 0.75$	33.54	31.66	32.55	28.76
Quadruped $\beta = 0.85$	39.48	43.91	42.54	21.73
Hexapod $\beta = 0.5$	67.69	68.71	68.25	60.31
Hexapod $\beta = 0.8$	86.44	81.59	84.02	77.39
Hexapod $\beta = 0.8$ (uneven terrain)	93.13	n/a	n/a	78.33
Quadruped $\beta = 0.8$ (w 0.02 rad slope)	52.15	n/a	n/a	20.58
Real Quadruped $\beta = 0.85$ (w/uneven weight)	52.5	n/a	n/a	47.28

Table 2. Measured SSM for hexapod and quadruped

A similar response for the three PCRG algorithms can be appreciated. The addition of the control loop increased noticeably the robot stability margin. Also, for higher support factors the SSM increased as should be expected in the geometric model. It must be noticed that replacing the FFNN in the previous model, by the parametric description of the leg trajectory, the synthesized walking patterns do not exhibit any undesired vibration. For simulated and real conditions, the quadruped robot was able to walk over a terrain with a low slope in a case, and with uneven weight distribution for the other. In both cases the measured SSM was improved by using BFP reference generator.

## 6. Conclusions and Future Works

### 6.1 Conclusions

A state of the art review was exposed for locomotion modes in quadrupeds and hexapods. In the review were identified the most relevant components for each neurophysiologic model; also the advantages and disadvantages of each model were discussed. It must be noticed that some coincidences in the proposed problem, related to the modeling using not only the conventional method but also the neurophysiologic approach were found; in both cases, the model is based on two systems: one modeling the temporal coordination among the legs and the other one modelling the trajectory control for each leg. The proposed idea is to divide the locomotion trajectory generation issue in two problems: the coordination of the phase relationships among the legs and the controlled movement of the joints for each leg, simplifying the design and implementation for the whole locomotion system.

One of the models presented was a locomotion model based on Recurrent Neural Networks (CTRNN), synthesized using genetic algorithms. The locomotion system is based on CPG concept, using coupled oscillators and NN. In order to analyze the output waveform of the temporal trajectory of the legs, a fitness function was employed. Such model leads to an explicit control of the leg speeds during the locomotion, and to control also the support factor, to control the phase relationships among the legs and also to the explicit control of the spatial trajectory described by each tip of the legs. It must be pointed out that the parameter synthesis of the CTRNN using GAs does not assure the absolute convergence to a practical solution.

The feedforward neural networks were used in two different applications: one, in the determination of the transition profiles during the movement of one leg; the other, for the transformation of temporal references into spatial references. With the use of feedforward neural networks it was possible to get a model for the locomotion trajectories whose main



structure is independent of the kinematics model of the robot leg. The use of the model directed to get soft transitions among different spatial trajectories of the walking profiles for the 3 DOF legs of a quadruped robot. It has been shown that it is possible to synthesize the desired trajectories for 3DOF quadruped legs using simple feedforward neural networks. It is reasonably expected that this method could be extended to other kind of walking machines after doing the proper modifications of the method.

The problem of the modeling of the locomotion system using ACPO was solved using a feedforward neural network connected to the output of the vector states of the coupled oscillators. It must be noticed that the problem of coordination of the movement of one leg using ACPO had not been solved to the present. Coupled oscillators issue with magnitude changes due to gait mode variations was solved by employing only the phase information of the output vector.

The problem of margin stability arises for the platform control. To improve the SSM, platform accelerations and ground contact measurements were taken during online operation of robotic platform. It was observed the effects of overfitting in the training of the neural network. Such overfitting produced low amplitude oscillations during walking phase. This is closely related to the number of neuron units in the hidden layer. Special care in this issue is recommended to avoid stability problems in higher speed walking modes. Also it was pointed out the effect that can have neural network on support factor, reducing it due to waveform approximation task. It is suggested to study other neuron function kernels in order to reduce this problem. This parameter, the support factor, is employed in the conventional locomotive geometric model. The parameter is represented here through a companding curve of the phase for the temporal reference of each leg, being completely independent of leg kinematics and specific implementation of temporal subsystem.

By including additional control inputs to the network, it could be possible to achieve a higher level control for robot platform variables, like body inclination and weight distribution by the use of accelerometers and ground contact sensors.

## 6.2 Future works

It is mandatory to review different training methods for the RNN employed to model the locomotion system. Using genetic algorithms it was shown that convergence is not assured. The training methods must use as training samples the spatial trajectories of the joints of each leg of the quadruped. Also, it must be emphasized the feasibility to control the phase relationships among the networks that control each leg of the robot. The problem observed of overfitting in the training stage of the NN must be studied in dependence with the neuron number and the structure of the hidden layer and its influence on stability, vibrations and support factor of the platform.

It must be studied the viability to implement the generation of spatial trajectories through coupled differential equations like the ones employed in ACPO. Such implementation must be oriented to generate an attractor space where the state vectors converge to the desired spatial trajectory in order to control each leg. It is relevant to be capable to control the final trajectory of the system with dependence of the parameters employed in the geometric locomotion model.

It is needed to study the impact of the variations of magnitude in state vectors of the ACPO during the walking modes transition. Normalization of such vectors or the control of its magnitude during the companding phase must be granted. In this way it could be reduced

the time that the trajectory remains in space points that do not belong to the trajectory training examples of the NN.

A variant of the generator proposed based on ACPO, could be studied the performance of the model employing the information provided by the magnitude and angle of the state vectors of the oscillators instead of its  $\{x, y\}$  components.

In the near future some different approaches are going to be tested, as combination of gait synthesis using the FFNN with strategies of position-force control on the quadruped leg. With this approach it should be possible to overcome more significant terrain irregularities and other external perturbations.

## 7. References and Bibliography

- Amari S. (1972). Characteristic of the random nets of analog neuron-like elements. *IEEE Transaction on System, Man and Cybernetics*, SMC-2:643–657.
- Ames J. (2003). Design methods for pattern generation circuits. *Master's thesis*, Case Western Reserve University.
- Arsenio A. M. (200). Tuning of neural oscillators for the design of rhythmic motions. *In ICRA*, pp 1888–1893.
- Bares J. E., Whittaker W. L. (1993). Configuration of autonomous walkers for extreme terrain. *The International Journal of Robotics Research*, 12(6):621–649.
- Billard A., Ijspeert A. J. (2000). Biologically inspired neural controllers for motor control in a *IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Volume VI, pages 637–641. IEEE Computer Society.
- Buchli J., Ijspeert A. J. (2004). Distributed central pattern generator model for robotics application based on phase sensitivity analysis. *First International Workshop, BioADIT 2004*, volume 3141 of Lecture Notes in Computer Science, pp 333–349.
- Buchli J., Righetti L., Ijspeert A. J. (2005). A dynamical systems approach to learning: a frequency-adaptive hopper robot. *In Proceedings of the VIIIth European Conference on Artificial Life ECAL 2005*, LNAI, pp 210–220. Springer Verlag.
- Cappelletto J., Estévez P., Medina W., Fermín L., Bogado J. M., Grieco J. C., Fernández G. (2006). Gait synthesis and modulation for quadruped robot locomotion using a simple feed-forward network. *In Lecture Notes in Computer Science - 4029*, pp 731–739. Springer Berlin / Heidelberg.
- Cappelletto J., Estevez P., Grieco J. C., Fernandez-Lopez G., Armada M. (2006). A CPG based model for gait synthesis in legged robot locomotion. *In Proceedings CLAWAR 2006*, pp 59–64, Brussels, Belgium.
- Cappelletto J. (2007). Gait generator for a quadruped robot using neurophysiologic principles. *Master's thesis*. University Simón Bolívar. Caracas, Venezuela.
- Cappelletto J., Estévez P., Fernandez-Lopez G., Grieco J. C. (2007) A CPG with force feedback for a statically stable quadruped gait. *To be published in CLAWAR 2007*
- Conradt J., Varshavskaya P. (2003). Distributed central pattern generator control. *Technical report*, ETH / University Zurich.
- Di Paolo E. (2002). Evolving robust robots using homeostatic oscillators. *Technical report*, University of Sussex.
- Franceschi H. (2005). Análisis, Diseño y Evaluación de Estrategias de Control de Fuerza en Robots Caminantes. *PhD Thesis*, Universidad Complutense de Madrid. Spain.

- Funahashi K. I. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks*, 2:978–982.
- Funahashi K., Nakamura Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6:801–806.
- Gallagher J. C., Chiel H., Beer R. D. (1999). Evolution and analysis of model cpgs for walking: I. dynamical modules. *Journal of Computational Neuroscience*, pp 99–118.
- Gallagher J. C., Chiel H., Beer R. D. (1999). Evolution and analysis of model cpgs for walking: II. general principles and individual variability. *Journal of Computational Neuroscience*, pp 119–147.
- Garcia E., Gonzalez de Santos P. (2003). Adaptive periodic straight forward/ backward gait of a quadruped. In *13th International Symposium on Measurement and Control in Robotics*, Madrid, Spain..
- Gerstner W., Kistler W. M. (2002). Spiking Neuron Models. Single Neurons, Populations, Plasticity. *Cambridge University Press*.
- Ghigliazza R. M., Holmes P. (2004). A minimal model of a central pattern generator and motoneurons for insect locomotion. *In Press*.
- Ghigliazza R. M., Holmes P. (2004). A minimal model of bursting neurons: The effects of multiple currents and timescales. *SIAM J. on Applied Dynamical Systems*, 3(4):636–670.
- Grieco J. C. (1997). Robots escaladores. Consideraciones acerca del diseño, estabilidad y estrategias de control. *PhD dissertation*, Universidad de Valladolid.
- Hodgkin A. L., Huxley A. F. (1952). A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. *Journal of Physiology*, 117:500–544.
- Ijspeert A. J., Kodjabachian J. (1999). Evolution and development of a central pattern generator for the swimming of a lamprey. *Artificial Life*, 5(3):247–269.
- Kimura H., Fukuoka Y., Hada Y., Takase K. (2002). Three-dimensional adaptive dynamic walking of a quadruped - rolling motion feedback to cpgs controlling pitching motion. *IEEE International Conference on Robotics and Automation, ICRA 2002.*, pp. 2228–2233, 2002.
- Kimura H., Fukuoka Y., Cohen A. (2003). Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *International Journal of Robotics Research*, 22(3-4):187–202.
- Kuo A.1 D. (2002). The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor Control*, 6:129–145, 2002.
- Lewis M. A. (2002). Perception driven robot locomotion. *Journal Robot Society of Japan*, pp 51–56.
- Marder E. (2001). Moving rythms. *Nature*, 410:755.
- Marder E., Bucher D., Schulz D., Taylor A. (2005). Invertebrate central pattern generator moves along. *Current Biology*, 15:685–699 (R).
- McGhee R. B., Frank A. A. (1968). On the stability properties of quadruped creeping gaits. *Mathematical Bioscience*, Vol 3. pp 331- 351.
- Molter C. (2004). Chaos in small recurrent neural networks : theoretical and practical studies. *Technical report*, Univ. Libre de Bruxelles.

- Nepomnyashchikh V. A., Podgornyj K. A. (2003). Emergence of adaptive searching rules from the dynamics of a simple nonlinear system. *Adaptive Behaviour*, 11(4):245-265.
- Paine R. W., Tani J. (2004). Evolved motor primitives and sequences in a hierarchical recurrent neural network. In *GECCO* (1), pp 603-614.
- Schmitz J. (2001). Biologically inspired controller for hexapod walking: Simple solutions by exploiting physical properties. *Biol. Bulletin*, pp 195-200.
- Simó L. S., Lewis M. A. (1999). Elegant stepping: A model of visually triggered gait adaptation. *Connection Science*.
- Solidum A., Lewis M. A., Fagg A. (1992). Genetic programming approach to the construction of a neural network for control of a walking robot. *IROS'92*, pp 2618-2623.
- Susuki R., Nishii J. (1994). Oscillatory network model which learns a rhythmic pattern of an external signal. In *IFAC Symposium*, pp 501-502.
- Talebi S., Poulakakis I., Papadopoulos E., Buehler M. (2000). Quadruped robot running with a bounding gait. In *ISER*, pp 281-289.
- Todd D. J. (1985). *Walking Machines. An Introduction To Legged Robotics*. Kogan Page Ltd., Pentonville Road, London N1 9JN.
- Tsung F. S. (1994). *Modeling Dynamical Systems with Recurrent Neural Networks. PhD Thesis*, Univ. of California, San Diego.
- Wettergreen D., Pangels H., Bares J. (1995). Behavior-based gait execution for the dante ii walking robot. *IEEE International Conference on Intelligent Robots and Systems (IROS)*.
- White H., Hornik K., Stinchcombe M. (1989). Multilayer feedforward networks are universal function approximators. *Neural Networks*, 2:359-366.



## **Bioinspiration and Robotics Walking and Climbing Robots**

Edited by Maki K. Habib

ISBN 978-3-902613-15-8

Hard cover, 544 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, September, 2007

**Published in print edition** September, 2007

Nature has always been a source of inspiration and ideas for the robotics community. New solutions and technologies are required and hence this book is coming out to address and deal with the main challenges facing walking and climbing robots, and contributes with innovative solutions, designs, technologies and techniques. This book reports on the state of the art research and development findings and results. The content of the book has been structured into 5 technical research sections with total of 30 chapters written by well recognized researchers worldwide.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

J. Cappelletto, P. Estévez, J. C. Grieco and W. Medina-Meléndez and G. Fernández-López (2007). Gait Synthesis in Legged Robot Locomotion Using a CPG-Based Model, *Bioinspiration and Robotics Walking and Climbing Robots*, Maki K. Habib (Ed.), ISBN: 978-3-902613-15-8, InTech, Available from: [http://www.intechopen.com/books/bioinspiration\\_and\\_robotics\\_walking\\_and\\_climbing\\_robots/gait\\_synthesis\\_in\\_legged\\_robot\\_locomotion\\_using\\_a\\_cpg-based\\_model](http://www.intechopen.com/books/bioinspiration_and_robotics_walking_and_climbing_robots/gait_synthesis_in_legged_robot_locomotion_using_a_cpg-based_model)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.