Goal-Based Control and Planning in Biped Locomotion Using Computational Intelligence Methods

Juan José Figueredo Uribe

Tutor:
Jonatan Gómez Perdomo, PhD

Submitted in total fulfilment of the requirements
of the degree of Master of Engineering

December 15, 2008

Facultad de Ingeniería
Universidad Nacional de Colombia

DRAFT

Contents

Heaven's just a scab a away.
I'd like to see you after just one taste.

The Mars Volta (2001-)

1 Introduction

1.1 Goals

Dynamical bipedal walking has been a key objective in robotics since its origins, due to the human curiosity about artificial anthropomorphic beings which gave rise to the robot concept itself [7]. As could be seen in most industrialized countries, the industrial manipulators have found a wide adoption, and there is little space to a major boost in the area [49]. Although, the interest is shifting from an industrial point of view to a more domestic one [1], where robots can be seen as additional aids to human daily tasks.

But, in order to accompany humans, the robot must be able to fluidly move through all the environments in which the human can, and those environments are devised to adapt well to anthropomorphic beings: factories, vehicles, houses, sidewalks, and shopping malls, among others. This way, a robot made to perform well in arbitrary environments will have a great advantage if it is anthropomorphic, so that it could serve well as an personal assistant [11].

The interest on biped robotics is not only for biorobotics itself. Another reason to research anthropomorphic motion is the understanding of human morphology, mechanics and control, from a medical point of view, where robotics could serve as a testing scenario to both theories and technologies concerning human motion (for an example see [43]) and, probably, provide technological aids and substitutes to body parts when an impairment is present [17].

Another motivation to the research of biped walking is related to the fact that anthropomorphic motion planning and control is a complex problem that includes nonlinear and non-holonomic systems [3], complex computing tasks, adaptability to unknown and unstructured environments [10], among others, and is useful to test different mechanical, electronic, computing and control techniques applicable to diverse areas. As it is remarked by Craig [**?**], it should not be forgot that the predominant dynamics algorithm for open-chain mechanisms was developed [**?**] and refined [**?**] while working in biped walking problems.

Among the different problems faced in anthropomorphic motion, biped walking is one of the most difficult because its intrinsic instability. In contradistinction to wheeled mobile robotics and stable legged robotics, biped robotics must allow locally instable motion in order to attain a fluid locomotion. Additionally, there are several problems in biped locomotion other than stability. A bipedal walker must be in capacity of choose the best path to reach an objective, avoid obstacles, tolerate high perturbations, perform well in unstructured environments, and move with an optimal energy consumption.

This way, while already a problem broadly studied, the problem of biped walking is still open for several key goals not currently attained, such as optimal energy consumption, objective-based planning and walking in less structured environments. However, there are recent, and not so recent, notable contributions in this field given by diverse methods: active control, passive dynamic walking, and computational intelligence; which are giving a stronger basis for major advances in the field, fueled by a strong academic and commercial motivation.

Nonetheless, here we study a narrower problem, in which biped robotics combines with dynamical neural neurons and evolutionary robotics to pursue biped walking control by emergence, i.e. without the explicit specification those parameters required in order to achieve the desired behavior with the control scheme chosen.

In a intuitive way, the problem studied can be broadly described as to provide a biped robot with an optimal locomotion. Nonetheless, since the optimality referred can be thought as a performance measure, it is convenient to be more specific in its definition. Based on the preliminaries in previous sections, the relevant goals to be accomplished by biped locomotion proposed are:

- To conserve static stability: The ability to conserve equilibrium while in static state or in quasi-static motion.
- To walk conserving dynamic stability: The ability to walk, or locomote without leaving the contact to the floor, at velocities high enough to make considerable the inertial forces maintaining stability. That means the robot should walk indefinitely without falling if there are no obstacle in its way.
- To locally minimize energy consumption: For a given path minimize the energy consumption of the gait pattern.
- To react to external perturbations: Sense and compensate external perturbations that affect locomotion, including external forces and changes in environment parameters.
- To plan gait trajectories to attain specific objectives: The ability to choose a suitable trajectory to reach a desired state, not only in terms of path, but also joint coordination.
- To move across unstructured environments: Perform biped locomotion in environment with variable conditions which are not previously given to the robot, and plan according to perception of those environment variable conditions.

- To globally minimize energy consumption: For a given objective in an arbitrary environment, select and perform the gait trajectory that minimizes energy consumption.

Although non strictly, the order of goals presented has in mind a growing complexity required to satisfy them. That implies that in some cases it would be useful to satisfy, at least partially, a previous one before the one following it.

The last goal proposed, the global minimization of energy consumption, has not been satisfied by any of the works reviewed so far, and the goals of motion across unstructured environments and planning of trajectories to attain specific objectives have been taken into account by few of them . Particularly, in the area of computational intelligence it is necessary first to address consistently the problem of planning previous to studying the remaining goals.

Therefore, the specific problem defined is to perform the motion planning and control of biped locomotion in a computer-simulated environment using a goal-oriented integrated architecture based on computational intelligence. The biped model defined is a rigid body linked model, with the lesser number of joints required to move in a two-dimensional environment.

The motion planning is understood as the process to detail a motion task into a sequence of reference actions, while the motion control is the process that transforms a sequence of reference actions into control signals applied to robot actuators (applying torque accordingly to each joint).The computer-simulated environment is essentially a physics engine in which the dynamical experiments can be performed, given a model of the system, environmental conditions and a set of inputs.

1.2 Main Contributions

The goal-oriented integrated architecture proposed is the main contribution of the thesis proposal, since in the work reviewed there are no goal-oriented motion planning strategies applied to biped locomotion, and therefore there are no planning and control integration proposals. In order to be able to move consistently across an unstructured environment and to be useful for any specific task it is required that the biped can pursue specific goals. The integration of planning and control has two purposes. The first one is to allow a uniform and hierarchical implementation across the architecture, allowing the same adaptation strategies and analysis tools to be applied to both planning and control systems, and performing the planning task dynamically in the same time scale in which the control system is working. The second one is to be able to change softly from one operation mode to another when goals have changed, controlling undesired high frequency components of the reference control signal.

The proposed problem will face dynamic biped walking in structured environments without perturbations. With the contribution in goal-oriented planning and its integration to motion control using a computational intelligence approach, it is expected that there will be a framework such that the motion across unstructured environments and the global energy minimization fall in the field of study in short time.

1.3 Contents Overview

Sink your teeth into flesh of midnight.
Night forever more.
The Mars Volta (2001-)

2 Preliminaries

2.1 Introduction

Dynamical bipedal walking has been a key objective in robotics since it was first conceived, due to the initial idea of robots as anthropomorphic machines with human-mimicking behavior [7]. In spite of the advent of industrial robotics and a modern academic and industrial conception more oriented towards robotics as a production augmenting value [49], there are not only important reasons to pursue anthropomorphic robots and understanding of bipedal walking, but an increasing interest on non-industrial robots (for which the term service robot is sometimes used) [1].

Most of the environments existing are devised to adapt well to humans: factories, vehicles, houses, sidewalks, and shopping malls, among others. This way, a robot made to perform well in arbitrary environments will have a great advantage if it is anthropomorphic, and therefore it could serve well as an personal assistant [11]. Another reason to research anthropomorphic motion is the understanding of human morphology, mechanics and control, from a medical point of view, where robotics could serve as a testing scenario to both theories and technologies concerning human motion (for an example see [43]) and, probably, provide technological aids and substitutes to body parts when an impairment is present [17]. A third motivation is related to the fact that anthropomorphic motion planning and control is a complex problem that includes nonlinear and non-holonomic systems [3], complex computing tasks, adaptability to unknown and unstructured environments [10], among others, and is useful to test different mechanical, electronic, computing and control techniques applicable to diverse areas.

Among the different problems faced in anthropomorphic motion, biped walking is one of the most difficult because its intrinsic instability. In contradistinction to wheeled mobile robotics and stable legged robotics, biped robotics must allow locally instable motion in order to attain a fluid locomotion. Additionally, there are several problems in biped locomotion other than stability. A bipedal walker must be in capacity of choose the best path to reach an objective, avoid obstacles, tolerate high perturbations, perform well in unstructured environments, and move with an optimal energy consumption.

2.2 Biped Walking

Biped walking is, from a robotics standpoint, the articulated motion of several bodies with motion between them coordinated by actuation torques in their joints. It is easier to model biped walking as a rigid bodies system, but the nature of biological tissues is best modeled by viscoelastic bodies. Also, it is easier to model biped walking with the most reduced number of degrees of freedom (i.e. the minimum number of independent parameters required to completely describe the sate of the system), but at the organ level there are at least 6-DOF in each leg (in the hip: Flexion-Extension, Abduction-Adduction, External-Internal Rotation; in the knee: Flexion-Extension; and in the ankle: Plantarflexion-Dorsiflexion, Pronation-Supination) for a total of 12-DOF in the lower extremities. Though, a biologically complete description of joint states would need a lot more degrees of freedom: there are about 650 skeletal muscles in human body, each one composed of sev eral muscle fibers (myocytes) independently innervated. This way proper selection of model body and joint properties is crucial in modeling walking. Biped walking

2.2.1 Static and Dynamic Stability

One important element in biped walking is stability. Generally speaking, there are two ways of defining stability in biped locomotion. The first method, assumes locomotion as a quasi-static motion and, supposing negligible inertial momentum in body links, the stability is given by the location of body center of mass (CoM). If it is inside of the convex hull containing all points in the support area (i.e. the supporting polygon) it is said to be stable, otherwise it is said to be unstable. An example of a statically stable biped robot is shown in Fig. **??**. This principle applies generally under low accelerations in the single-support phase, because the support polygon in this phase is limited to the contact area of the supporting feet, and therefore the robot has a very limited movement range in order to conserve stability. In the double-support feet it can be used reasonably to model stability under higher accelerations, because the support area is wider and the stability is less sensible to dynamic forces.

The second method, takes in account the inertial forces due to dynamical motion, and its general definition is still and open problem [2]. However, a limited way to determine biped dynamic equilibrium is given by the projection of the dynamical equivalent to CoM: the Zero Moment Point (ZMP). The Zero Moment Point is the point where the reaction force with the supporting element would produce a zero moment. If the ZMP is located inside the supporting polygon, unstabilizing moments due to dynamic locomotion would be compensated by the support reaction. An illustration of a robot which is not in equilibrium is shown in Fig. **??**. The ZMP is the projection over the floor of the total force acting over the CoM. When accelerations become relevant, the reference frame placed in the CoM is not anymore an inertial frame and, therefore, appears an inertial force opposed to CoM acceleration, which, with the weight force, yield a total force not normal to the floor. In order to be in moment equilibrium, the projection over the floor of the total force must coincide with the floor reaction point, that is, the point where the equivalent reaction force is applied.

Biped locomotion can be thought as a rhythmic and symmetric progression of trunk and limbs movement. In a complete gait cycle each leg passes trough a stance phase, in which its toe is in floor contact, and a swing phase, in which the foot is balancing without floor contact. Between them is a double support phase, where both feet are conforming the support area. Double support phase dynamics is more complex than that of the single support because of the additional constraints that yields a parallel kinematic chain. Thereby, a usual modeling technique is to model the dynamics of single support and a punctual transition condition for the double support [13]. However, double support dynamics modeling could be representative since it accounts for 20% of the gait cycle.

2.2.2 Biological Considerations on Biped Locomotion

In order to build suitable controllers for biped locomotion it is necessary to define what aspects, or features, are most important to good performance, robustness, stability, adaptability, and optimality of walking. All of these properties are attained by biological biped walking and therefore it is the main inspiration for building models as well as controllers of locomotion.

There are several models of biped walking, and each of them remarks one or more important features. A very good review and analysis of biped locomotion models can be found in [39]. Vaughan reviews six models of bipedal walking: Bipedal walking as an evolutionary adaptation of hominids, minimization of energy consumption by displacing the CoM along an optimal path, progressive learning with risk of falling minimization, spinal cord interneurons acting as rhythmic central pattern generators, neural system training along with biomechanical system and environment adaptation, and feedback control in powered dynamic locomotion. The neurobiological aspects of motion control have been examined in [12]. Duysens remarks the reflex response of sensor inputs in motion control and presents tree levels for neural motion control: Feedback control in motoneurons, in terms of contribution of reflex action over motoneuron signal intensity; feedback control in central pattern generator flexor-extensor centers, relevant to movement synchronization and reflex response to perturbation and loads; and higher level control, referred to conscious control of locomotion. Duysens states that despite of the relative low understanding of the high level control participation in locomotion, the two lower levels are complex and rich enough to be studied an applied to robot design (i.e. the so-called *spinal robot*, because it would have the control architecture expected in a human with transected spinal cord).

2.3 Neural Fields

Neural fields arise as a tissue level model of neural populations in brain. It has been proposed by Wilson and Cowan [?] and detailed by Amari [?] in the particular case of lateral inhibition. In this model, neural population in considered continuum in which exists a dynamical evolution equation where the mean activation potential evaluated in one place is affected by its neighborhood according to a so-called mexican hat function (as noted by Coombes [?] better called wizard hat function) in which close neighbors act as exciters and distant ones act as inhibitors.

The base model, as presented by Amari [?] for the multiple layer case is:

$$\tau_i \frac{\delta u_i(x,t)}{\delta t} = -u_i + \sum_{j=1}^{m} \int w_{i,j}(x,x';t-t') f_j\left(u_j(x',t')\right) dx'dt' + h_i + s_i(x,t) \tag{2.1}$$

Where τ is a temporal constant of synaptic decay rate, $u_i(x,t)$ is the average membrane potential of the neurons located at position x at time t on layer i (where x can be 1-dimensional, 2-dimensional or even of higher dimension). The average intensity of connection from neurons on layer j at y to neurons on layer i at x is modeled with $w_{i,j}(x,y)$, $f_j(\cdot)$ is the saturating output function which is monotonically nondecreasing. The deviation of the average stimulation potential at place x at time y of layer i is represented by $s_i(x,t)$, and $h_i = \bar{s_i} - r_i$ is the sum of the average stimulation potential an the resting potential of layer i.

There are several assumptions that produce simplifications over the previous model. One of them is to include the additional dependence of the time lag of signals $t' = |x - x'|/v$ where v is the velocity of an action potential [?]. Nonetheless, while not stated otherwise we will not take into account the time lag, as well as the multiple layers. We will also merge the non-homogeneous terms $S(x,t) = h + s(x,t)$. This way, the resulting equation takes the form (for x n-dimensional):

$$\tau \frac{\delta u(x,t)}{\delta t} = -u + \int_{\mathbb{R}^n} w(x,x') f\left(u(x')\right) dx' + S(x,t) \tag{2.2}$$

For further simplification, temporarily we will consider the connection kernel $w(x,x')$ as isotropic and homogeneous, so that it only depends on the norm of the vector difference $\|x - x'\|$ i.e. $w\left(\|x - x'\|\right)$. Amari found diverse stable-state solutions for the one-dimensional case (isotropic and homogeneous), where the model is:

$$\tau \frac{\delta u(x,t)}{\delta t} = -u + \int_{-\infty}^{\infty} w\left(|x - x'|\right) f\left(u(x')\right) dx' + S(x,t) \tag{2.3}$$

The typical form for the connection kernel (for which Amari obtained his results) can be seen in figure 2.1.

Figure 2.1: Wizard hat function used as connection kernel.

Also, the typical sigmoid saturation function, as well as its maximum gain limit case (the Heaviside function), are shown in the figure 2.2.

Figure 2.2: Sigmoid and Heaviside saturation functions.

2.4 Evolution and Adaptation

2.4.1 Evolutionary Algorithms

Evolutionary algorithms are a set of population-based heuristic search and optimization techniques. They maintain a population, and apply a set of operators or transformations over its members. Those operators are typically inspired on biological evolution and usually include selection, reproduction and mutation, among others. The operators are dependent of the evaluation of a performance function called fitness function. Generally, fitness function evaluation may include, from a simple numerical evaluation, to a complex simulation, in order to get the performance criterion which its optimization is pursued.

The pseudo-code of a general evolutionary algorithm is as follows:

Algorithm 1 *EvolutionaryAlgorithm*

```
P ← Generate initial population of size N
Evaluate fitness for each individual in P
repeat
    P′ ← Apply operators to P
    Evaluate fitness for each individual in P′
    P ← Select N individuals in P′ according to a selection scheme
until Termination condition is met
```

The most predominant form of an evolutionary algorithm is embodied by genetic algorithms. They most frequent genotypical representation is a bit sequence, although other representations can be used. Usually they are implemented with a generational replacement of population, but in some situations it is useful to conserve a small set of the better individuals across generations in a steady-steady replacement.

2.5 Computational Intelligence Applied to Biped Robotics: A Survey

Here is presented a tentative taxonomy of the previous works made in the area of motion planning and control methods in biped walking. Three major approaches are identified: Computational Intelligence, Active Control and Passive Dynamic Walking. Fig. **??** shows for a graphical representation of the taxonomy with emphasis in computational intelligence methods.

Before focusing on computational intelligence methods, the two other methods mentioned are briefly presented.

Active Control refers to the persistent control of the joint actuation and state applying control theory. It includes position, velocity, acceleration and torque control techniques. Also, several design methods can be used, including some derived from linear ones but applied to nonlinear systems. In the methods used are included methods based on frequency response, performing the optimization of a performance criterion (e. g. H_∞ control), or applying state feedback. Another simpler method used is the tuning of parameters of a PID controller. Also it is possible to a given extent use local linear approximations and directly use linear control methods. Active control gives the best trajectory tracking and accuracy, provides methods to evaluate stability (such as Lyapunov stability analysis), and can be vertically integrated to higher motion planning algorithms, but its main deficiency is its very high energy consumption due to the persistent control of joint actuation, even when optimal control is applied.

The other major approach is Passive Dynamic Walking (PDW). In PDW it is took the opposite approach by totally suppressing any joint actuation. The idea is that the actuation given by gravity force over a biped robot standing over a slightly inclined surface, in conjunction with its natural dynamics, must achieve stable gait patterns. This is attained by using passive elements as springs and dampers and a careful mechanical design guided by a detailed analysis by dynamic systems theory, including state space modeling, phase transitions, and probably other techniques such as PoincarÃ¨ maps and Lyapunov stability analysis and, mostly, dynamic simulation. In order to obtain gait pattern in flat surfaces it can be applied reinforcement learning in a control system so as to learn how to simulate the slight actuation given by the gravity in the inclined surface case. The main advantage of this method is its very low energy consumption, and its main disadvantage is its low flexibility to track general paths, particularly those including vertical movements, and its high sensibility to environment conditions.

Next, each one of the three major approaches to the problem that apply computational intelligence are examined. These are Central Pattern Generator methods, Dynamic Walking Control methods, Static Walking Control methods, and Trajectory Tracking methods. This classification emphasizes the way in which the problem is solved and not the subfield of computational intelligence used.

2.5.1 Central Pattern Generator (CPG) Methods

A Central Pattern Generator (CPG) is a system which is supposed to give the coordinated rhythmic stimulation to joints required to generate a gait pattern. It is inspired from the spinal motor center find in animals such as mammals, and it is usually implemented using a kind of neural networks called oscillatory neural networks.

Neural oscillators are a special type of artificial neural networks (ANNs) described by two essential elements: the dynamical properties of the individual neuron, and the type of coupling between neurons. Each neuron has its dynamics described by a set of differential equations.

A more specific model of neural oscillators couples neurons in pairs. It is based in the fact that most muscle fibers have one excitatory center for flexion and another for extension, and them are inhibiting between them. This way, the two coupled neurons can be represented by a set of differential equations. An example of this arrangement is the Amari-Hopfield model, for which a grafical representation is shown in Fig. presented here as shown in [30]:

$$\tau \dot{u} = -u + af(u) - cf(v) + S_u(t) \tag{2.4}$$

$$\tau \dot{v} = -v + bf(u) - df(v) + S_v(t) \tag{2.5}$$

In which u and v are the neuron potentials, $S_u(t)$ and $S_v(t)$ are the respective neuron inputs as function of time, and $f(u)$ and $f(v)$ are neuron outputs after applying the transfer function:

$$f(x) = \frac{1 + tanh(\mu x)}{2} \tag{2.6}$$

The parameters a, b, c and d characterize the neural oscillator behavior. This kind of arrangement is typically followed in the CPG control methods, where each neural oscillator (neuron pair) is used to stimulate a single joint, and the mutual interaction of neural oscillators conform the CPG.

A classical author in the application of CPGs to motion control of biped locomotion following the methodology presented above is Taga, who in his preliminary work of 1991 (see [38]) applied the concept of coupled neural oscillators, remarking the self-organizing properties of the neural system with the physical system and the environment, which gives some disturbance rejection that keeps the equilibrium of biped walking.

In general terms, the neural oscillator can be thought not only as the coupling of two neurons, but more globally, the rate of neuron activation potential variation can be a function of the network inputs, its current activation potential and a special term called fatigue. Also the rate of variation of fatigue is function of the current fatigue and its current output. A general expression for the system of differential equations that rule the dynamics of a individual neuron is [6]:

$$\tau_r \dot{x_i} = -x_i + \sum_{j=1, j\neq i}^{n} a_{ij} y_j - b f_i + s_i \tag{2.7}$$

$$\tau_a \dot{f_i} = -f_i + y_i \tag{2.8}$$

$$y_i = H_1(x_i) x_i \tag{2.9}$$

$$H_1(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{2.10}$$

Here, x_i denotes the activation potential of the ith neuron, y_i its output, a_{ij} the connection weight from neuron j to neuron i, f_i is the fatigue strength, b is the coefficient of fatigue, s_i is a bias factor, and τ_r and τ_a are time constants.

With this model Cao and Kawamura used an oscillatory neural network as CPG to generate biped walking patterns. The net composition includes a single neuron for each articulation, and all neuron are connected between them (i. e. all the networks is a single coupled neural oscillator). The neuron activation function is given by the set of equations presented above. A connectivity matrix, with inhibitory and excitatory weights (composed by the different terms a_{ij}) was evolved using an genetic algorithm to obtain a gait pattern. The chromosome is a linear arrange of matrix elements in binary representation in the form:

$$< \quad \underbrace{b_0 b_1 \ldots b_7}_{a_{12}} \quad \underbrace{b_8 \ldots b_{15}}_{a_{13}} \quad \cdots\cdots \quad \underbrace{b_{440} \ldots b_{447}}_{a_{87}} \quad >$$

The fitness function used is dived in two partial qualitative functions and a third which evaluates the correctness of the gait, applying three genetic algorithms, one nested in another (authors called it hierarchical evolutionary algorithm). This way they were able to control eight joints and generate successful gait patterns.

Next is presented relevant previous works that mainly use Central Pattern Generators as motion strategy.

Kun and Miller [23] implemented a control for biped robot walking of a 10-DOF robot, for which they used a CPG that uses a heuristic to generate sequences, synchronizing the performance in the joints with the natural dynamics of the robot, and receiving correction parameters in the lateral and frontal motion given by CMAC neural networks as inputs. Additionally, a third CMAC network is trained to help control the robot in the double support phase. The natural dynamics of the robot is fundamental for the control system given its dependency on directional balance to stabilize the movement. They used temporary difference learning to train networks. This pioneer work is validated implementing it on a real biped robot.

Venkataraman [40] made a complete compilation of the important elements in the CPGs in the field of biology and the robotics, and proposes an alternative method for biped, quadruped and hexapod locomotion patterns generation. The method has relation with the general implementation using oscillatory neural networks, but unlike these, it employs a single linear patterns generator, with which the movements are generated in each extremity using filters of finite dimension (adjustments of phase lead and phase lag filters) to simulate the delays that produce the coordinated movement necessary for locomotion. The oscillator is modeled including elements of the Van der Pool oscillator to obtain a nonlinear system with a robust and stable focus. There are presented as result patterns for the gait types already mentioned, proving that they can be obtained with a simple architecture.

Benbrahim and Franklin [4] developed a CPG using CMAC neural networks and simultaneously applying reinforcement and supervised learning. They use a central network, the generator, aided by a set of peripheral control networks in parallel with observation networks, whose inputs are relevant parameters that act as gait restrictions, among them body posture and height of body mass center. Peripheral control networks act only if its correspondent observation network detects a improper behavior from central network for a specific restriction. Control system output is determined by a Gaussian function with median located in the CPG output and standard deviation that initially has a high level and reduces as solution converges. Dynamic control necessary to follow the reference provided by CPG is easily implemented with a set of PID controllers. In the real application, the control scheme was aided by a posture bang-bang control (i. e. on-off control). It was required a previous training in order to obtain suitable network training time. This novel approach allows to prevent that CPG errors cause a general gait failure, not only providing robustness but accelerating the learning process.

Hasegawa et al. [15] proposed a method for developing pattern generators useful for a biped 13-DOF robot walking on inclined surfaces. They solved the unconstrained optimization problem with a hybrid evolutionary algorithm with a lower layer that generates trajectory points (from a cubic spline) using evolutionary programming (EP), which are added in interpolated point sets, and those set are in time evolved using a genetic algorithm (GA), selecting the successful sets that minimize energy consumed by the robot. Due to that each point evaluation depends on the best individual in the GA layer, and also second layer individuals a sets of elements of the first layer, the algorithm has a co-evolutionary scheme. In the practical implementation made, the dynamic control is performed by PD controllers.

Reil and Husbands [36] used an evolutionary algorithm (EA) to optimize the parameters of a fully connected recurrent neural network. The model of each neuron is a dynamic model that attenuates its answer in stimulation absence, and depends on a set of weights, a bias factor and a dynamic time constant, with a dynamics similar to the model exposed previously and presented in [6]. The three parameters types for the totality of network neurons are represented linearly in the individuals genotype used in the evolutionary algorithm. The fitness function used favors movement that maximizes displacement in frontal direction and penalizes movement in vertical direction. The authors manage with such simple evaluation rule to generate satisfactory patterns of march without any sensory feedback. Later, they add a sensorial entrance of auditive type and, leaving the recurrent connections fixed, they evolve the recurrent neurons connection weights with the sensory input. This way, they attain that the robot navigate towards a source of sound emission. The neural controller generates reference points that are taken to forces in the motors by a PD torque control. An addition to the efficiency of the algorithm is the premature abandonment of marches that move strongly in vertical direction. They have achieved a relative low success rate in evolutions (close to 20%) and propose switching to a more traditional CPG method.

Miyashita et al. [29] recalled the conceptualization of previous authors like Taga, emphasizing the importance of the oscillating behavior in the nervous system in form of CPGs. The proposed controller this way is based on oscillating artificial neurons, or oscillators, which are connected mutually in pairs to obtain a periodic behavior, in a form similar to the Amari-Hopfield model presented previously. It is shown the evolution of neural network structure using genetic programming (GP), in terms of the interconnections between oscillators, supposing the internal structure of the oscillators fixed (i.e. the parameters a, b, c, d an τ). The GP implementation for solving the problem is made representing the local network from each oscillator like an S-expression, doing a co-evolution where a subpopulation for each oscillator is evolved, but the evaluation of the aptitude function as well as the application of genetic operators are applied over an oscillator group, each one of a different population. Results of walks generated with this method are displayed, until a maximum of 10 stable steps. This work, spite its relative success, has a very large computational overhead and gives a very small set of successful neural oscillators. Nonetheless, its approach is very illustrative, and somehow resembles the spinal control proposed in [12].

Paul [33] argued that, unlike the controllers that use a CPG whose connections between oscillators in left and right sides are connected, it is possible to obtain a satisfactory biped walking with two decoupled generators. Two cases area treated: the first one decouples control and sensor system, having each an independent CPG, without any connection to each other and with entrances of sensors strictly located in his partial side; while the second one decouples connections but allows common entrances to both global position sensors. Both controllers are evolved in parallel with genetic algorithms (GA) to three morphologic parameters. It is shown that for the first case successful biped walking is achieved in most of cases but there are problems to follow a straight line. In the second, it is shown that the CPGs decoupling with global entrances is an equally satisfactory technique that the entirely coupled case. This way, Paul gives a highly valuable approach, in which a lot simpler technique achieves similar results to those more complex traditionally used. Paul therefore hypothesizes that lower limb control in walking is inherently reactive and sensory-motor coordination based, and CPGs are used only to alter gait conditions acting only in the trunk and arms.

In the another work [34], Paul aimed to continue the work oriented to controller simplification for the biped walking based on neural networks, this time showing that independent simple networks for controlling each leg can generate a stable walk, using only forward connections (i. e. a feed-forward neural network) without hidden layer. The used networks only receive as entrance the contact with their respective leg and a common bias signal for both controller. Of numerous configurations evolved with genetic algorithms, only two obtained the satisfactory long walk, but in a remarkable form, both indefinitely turned out to be stable. It is used, like in previous works of Paul, coevolution of some few morphologic parameters. This is probably the simpler solution to this problem proposed to the time.

Nakanishi et al. [31] made a less traditional approach to the generation of patterns for biped walking with some similarities to the CPGs. The idea consists of designing oscillators represented as systems of nonlinear phase coupled differential equations, in such form that the fundamental element of dynamics is not the oscillation frequency of each element, but its phase relations in order to obtain a desired movement pattern. A local weighted regression (LWR) is proposed as a training scheme to adapt elements phase, as well as dynamic system global oscillation frequency. Additionally, it is applied the concept of phase reset at the moment of heel strike. It is shown that the phase reset principle is advantageous when disturbances in real robot walking appear. Also, it is argued that the phase controller is simpler to train than CPGs. They also affirm the presented controller superiority in front of a controller by finite automata. It is remarkable that this work, first, abstracts the concept behind CPGs and, second, employs the already biologically supported phase reset. A minor flaw of this method is its high sensibility to initial conditions.

Komatsu and Usul [22] showed the control for different biped locomotion types using Hybrid Central Pattern Generators (H-CPG). The total action of the H-CPG proposed is determined by the sum of the individual actions of each one of its components: A neural oscillator that generates the rhythmical patterns in form of torques, a support force controller that applies the Jacobian matrix to map forces from cartesian space to joint space, and a position controller that implements a PD control to maintain legs as vertically as possible. The vertical and horizontal movements are separately processed by the force controller. They shown that proposed method allows to vary form slow walking to rapid walking, and also walking and running in modified environments, particularly in slopes. Satisfactory results were obtained in simulations and real robot implementation, and a high movement versatility was attained by complementing traditional CPG techniques with force and posture control. The system complexity in not so high and is indeed feasible.

Computational intelligence methods used are:

- Oscillatory Neural Networks: As nuclear elements of CPGs.
- Evolutionary Optimization: For optimization and training of CPGs
- Rhythmical Dynamic Systems: An alternative to oscillatory neural networks for building CPGs.

2.5.2 Trajectory Tracking Methods

Methods included in this category largely vary in implementation, but its main methodology consists of generating a kinematic pattern or succession, in a way that the joint following of it yields a successful gait pattern. Although there are applied computational intelligence methods in generating the trajectory, this process is typically followed by a simple control method, such as a PID control. Developments under this approach are relatively recent and have strong foundation on evolutionary computation methods.

The work of Capi et al. [8] gave and approach to obtaining a stable biped walk with an optimal power consumption for a biped robot of prismatic joints. The control system is based on the principle of Zero Moment Point (ZMP). The control loop for center of gravity (of PD type) receives a reference of the center of gravity location of the robot, and generates a control signal that receives the ZMP control loop, along with the current ZMP location. In the same way, the ZMP control loop processes that information and outputs actuation values for the rotational joint in the feet and the prismatic joint in the thigh. A genetic algorithm with real codification is used (equivalent to an evolutionary strategy) to look for the reference trajectory that generates the minimum energy consumption. The results with the optimization for minimum torque applied and optimization for constant height of center of gravity are compared. The problem of optimization with restrictions is turned to a problem without restrictions to which penalties in the aptitude function are added when the conditions of restriction are failed to fulfill. This way, Capi et. al present a gait trajectory generation method for an unusual robot configuration, and also give a forward step to optimization of energy consumption using evolutionary computation.

Later, Yamasaki et al. [46] applied the evolution by genetic algorithms (GA) to design a controller for a biped walker with low torque consumption. For it they proposed the optimization of the gait sequences generated in two phases. The gait sequences in terms of speeds are described like a set of sinusoidal functions with amplitude and phases as parameters, dependent on the angular positions of both legs. The parameters described are optimized along with the functions global angular speed. A binary representation is used, applying cross and mutation operators. The algorithm is divided in two phases, first with a fitness function proportional to total distance walked by the robot during the simulation time (or before falling), and the second one proportional to the walked distance and inversely proportional to the used energy. Therefore, a sequential optimization is made in order to, first, obtain suitable gait trajectories, and second, minimize energy consumed. Nevertheless, the best trajectory obtained attained less than 2 second of walking before falling, and this should be regarded only as a partial success.

Garder and Howin [14] recently applied a hybrid method of genetic algorithms (GA) and hill climbing to evolve walking patterns for a biped robot with pneumatic actuation. In the problem faced by them, the movement of the robot is restricted to sagittal plane, as to it is of concern only the robot frontal advance. The genotypical codification used is binary, and the complete sequence is represented by an individual, divided in three mechanism positions with pause times between them. The chosen method consists of leaving the durations fixed and using a traditional genetic algorithm to evolve the three positions that characterize the sequence of movement by 8 generations, and later to make an optimization by hill climbing, leaving fixed the bits of position, and generation after generation optimizing from the most significant bits of pause times to less significant ones. Satisfactory locomotion in the form of synchronous jumps is obtained. The short run time of the algorithm allows to implement it in real time in the robot. The result obtained where facilitated by the low problem dimensionality and relative smoothness of fitness landscape, but the method can fail on more complex problems.

Contemporarily Yanase and Iba [48] presented an implementation of an evolutionary algorithm with user interaction to determine sequences of movement, as well as other implementations with specific functions of aptitude to reach certain objectives. In the first case, the optimization of a sequence of movement by interactive evolutionary computation appears (IEC), for which a set of alternatives for each picture of movement is generated (keyframes), being these evaluated by the user graphically and, applying evolutionary operators in successive iterations along with the evaluation mechanism, each one of the pictures that are to form the sequence is obtained. Later they show how specially designed fitness functions can be used, altogether with a dynamic simulator, to optimize the movement to seat or kicking a ball. Also, they make a brief reference to the implementation of an multiobjective genetic algorithm (MOGA) for the solution of such problems. This is a novel approach in the fitness dynamical user evaluation employed and also in the alternative objective based approach to evolution.

The application of computational intelligence methods in this approach is focused on:

- Evolutionary Computation: For optimization of trajectories under a performance criterion (such as energy consumption, stability or specific motion objectives).

2.5.3 Dynamic Walking Control

These are probably the most complex methods among those based on computational intelligence. They aim to obtain dynamic stable walking by a control based on various techniques. Among them are included neural networks, fuzzy systems and evolutionary algorithms.

Some concepts used in this section have been already presented, but it is useful to detail recurrent neural networks, which are an important and promising technique. A recurrent neural network is usually modeled as a fully connected network without layered structure [16], in which the output can be expressed in vectorial notation as a linear combination of neuron states:

$$y_{rnn}(n) = Cx_{rnn} \tag{2.11}$$

And neuron states in $n+1$ are a function of recurrent potentials and input potentials in n, each of them multiplied by weight matrices.

$$x_{rnn}(n+1) = \varphi(W_a x_{rnn}(n) + W_b u_{rnn}(n)) \tag{2.12}$$

The function $\varphi()$ is a diagonal activation function, usually of sigmoidal shape. This way, a recurrent neural network is a dynamical system determined by its weight matrices W_a and W_a, its activation function $\varphi()$, and the linear combination of states used as output C. The relevant capability so obtained is that recurrent neural networks not only are a nonlinear mapping from inputs into outputs, but they present memory and dynamical response to inputs, which can give birth to emergent behaviors.

Next, some works in the field of dynamically stable walking are presented, beginning with the preliminary work of Magdalena and Monasterio-Huelin using fuzzy systems.

The work of Magdalena and Monasterio-Huelin [27] dealt with the evolution by genetic algorithms of a fuzzy logic controller and presents its application to locomotion patterns generation in a biped walker. Particularly, a genetic algorithm with binary codification is used to evolve the fuzzy rules as much as the ranks of normalization for each one of the variables involved, although all relevant information is codified to knowledge base, including the number of fuzzy sets by input and output variable and the definition of membership function functions. It is applied crossing between fuzzy rules, rule mutation at bit level and normalization rank mutation. The exposed methodology is used to design a fuzzy controller for a biped walker, which considers the different walking phases. A set of successful biped walks is obtained as result, and also there are obtained rules with better performance than the ones initially provided to the system.

The work of Bergener et al. [5] described an architecture that allows generating patterns of behavior as much as to dynamically control the execution of each task. Unlike other evaluated works, the architecture is applied to an anthropomorphic robot that is not biped and whose similarity with the human morphology is observed mainly in its arm, and therefore the problem faced do not include biped walking. The displayed architecture uses neural fields that map from sensor space to actuator space using principles of dynamic systems whose behavior adapts to conditions of the surroundings through bifurcation phenomena which respond to the so-called instanced dynamics (an elementary behavior parameterized by behavior variables in a specific environment). Also, to generate complex behaviors it is used a competitive dynamic system as arbiter mechanism, which includes a set of parameters that describes the logical and temporal requirements.

In the work of Capi citeCapi01Application the methodology to optimize the biped walk using genetic algorithms followed a procedure very similar to the one shown in [8]. It is observed as difference the application to a robot of rotational joints with 5 degrees of freedom (5-DOF), as well as obtaining both stable biped walking and stable ascent of stairs. Additionally, the data sets generated are used with the genetic algorithm to train a radial-basis neural network (RBFNN) with an hidden layer, of such form that locomotion patterns could be generated in real time.

In the novel work of Paul and Bongard [35] it is proposed a coupled evolution of morphology and control, since they emphasized the relative little depth with which the morphologic evolution (and in general the morphologic configuration) for obtaining gait patterns has been studied. The methodology applied consists of evolving simultaneously the morphology, in terms of the distribution of discrete mass blocks in a fixed joint configuration, and the control, in terms of the weights of a recurrent neural network. They made several experiments, varying the mass proportion that can be redistributed, with the purpose of evaluating the morphologic modifications in micro, meso and macro scales.

Juang [21] presented a method of trajectories generation using a feed-forward neural network as a nonlinear mapping between the present cinematic state and the control signal. It is based on the principle of periodic of the human walking, looking for to train the network of such form that diminishes the difference between the wished state and the end state, applying recurrent averaging learning in which the experiment is repeated consecutively initiating and finalizing in states determined by the average of the previous beginnings and ending, so that robustness in the system is obtained. The network is dynamically trained using backpropagation trough time.

Juang [20] also made an application of a neural control system for biped walking in slopes. The control system consists of: a neural controller incarnated by a feedforward neural network that generates torque signals in the actuators, a neural estimator that also consist of a feedforward neural network, and a third network with the same topology that adds a compensation control signal when there is an inclination in the surface. The method of training used is delayed backpropagation. The first step made is the identification of the system with the neural estimator. Later, the neural controller trains so that it follows a predesigned trajectory on flat surfaces, using the estimator to propagate the return error towards the control network. Finally, the two previous networks are left fixed and the inclination compensator, coupled with the control system, is trained using the estimator to propagate the error backwards. The obtained trajectory tracking, even in slopes, is satisfactory.

Wu et al. [44] studied the problem of inverted pendulum control excited in the base with two rotational degrees of freedom and non-acted movement in the base. There is a relation of such problem with the control of the trunk in biped walking, where the trunk is modeled as an inverted pendulum with such configuration. In order to obtain its objective, they use a set of feedforward neural networks with a hidden layer, which generate different functions that are connected in a simple closed loop configuration. An additional neural network is used to inversely model the system of such form that it is not required to measure the base position and an estimate is used instead. The controllers are pre-trained offline but its adjustment is dynamically made online providing adaptability to the controller. It is shown that the proposed control system evolves better than novel systems designed by control theory methodologies and in addition it does not require a model of the system nor a direct measurement of the position of the base.

Previously, un the same sense of previous work made by them, Wu el al. [45] studied the problem of inverted pendulum control with angular excitation in the base and free base translation in the three-dimensional space. They used then four neural networks to inversely model the system. Also a feedforward control with one neural network is used, and its output is feed back to the circuit. The inverse model is used to train the controller network.

Zhou [50] developed a learning agent with fuzzy and reinforcement learning (GAFLR). He had previously conceptualized several versions of agent GAFLR and its application the control of the biped long walk of a robot. The basic characteristics of the agent are: 1, Parts of a fuzzy knowledge base designed by an expert; 2. It is updated dynamically (online) using as reinforcement signal some parameter calculated with the fuzzyfication of system state measurements and internal estimations of future reinforcement signals; 3. The estimation of fuzzy reinforcement signals is made by a neuro-fuzzy network of 5 layers that is trained by reinforcement using a temporal technique of difference (temporal differences); 4. The actions are suggested by a neuro-fuzzy network of 5 layers trained with a genetic algorithm whose genotypic representation consists of the fuzzy rules that will be used for train the network directly; 5. A stochastic modifier of actions takes the composed reinforcement signal as the suggested action and generates the output. A fast convergence is exhibited towards the successful walk of the robot.

Zhou [51] developed also a simpler version of GAFLR agent, but without learning by genetic algorithms. There are shown the previous conceptualization and several versions of FLR agent and its application to robot biped walking control. Basic agent characteristics are shared with GAFLR version having discounted the training with GAs (the training by temporary differences is conserved). The operation of the stochastic action modifier module (SAM) is exposed better as an actions generator with normal distribution with average in FLR agent output, and standard deviation given by the reinforcement signal of previous iteration.

Park [32] applied the conceptualization of gait viewed from the zero moment point (ZMP), looking for, instead of generating movements for each one of the joints of the biped walker, dynamically generating the trajectories of the ZMP according to the hip joints state and the leg in balance, using a fuzzy logic system for trunk trajectory generation. Thus, it generates a natural movement and with reduced hip displacements. In order to calculate the movements of the joints once found the location of the ZMP, the inverse kinematics of the mechanism are used and a control by computed torque is applied. Leg trajectories are input to the trajectory generation system.

The hybrid algorithm proposed by Liu et al. [26] implements a controller for the double support phase in biped walking divided in three components. First it uses a CMAC neural network for which the fuzzy sets are generated, identified as generalization sets of the network, with which the input variables are mapped to a generalization space, obtaining the output as a linear combination between the representation generalized for the set of entrances and the matrix of weights of the CMAC neural fuzzy network. Later a hybrid position/force model is derived for the system including restrictions, with which later a model for the hybrid system is found applying a control scheme that includes the use of the neural fuzzy network. It was implemented an H_∞ optimization parameter for the joint model, which is optimized. Finally a robust hybrid controller was obtained which uses the neural fuzzy netowrks for the control of the inverse system and control of variable structure. Therefore, a robust method for desired trajectory tacking in double support is obtained. The study of techniques of switching system theory is suggested for, connecting them with the proposed control technique, integrally controlling the several gait phases.

Yamasaki, Nomura and Sato [47] emphasized the importance of dynamical phase modification (phase reset) for gaining stability in the biped walk, which is to be applied in the central patterns generator (CPG) implemented as a neural network, due to biological motivations. For it they approach two types of mechanical systems: first, a double pendulum, and second a 5-links biped walker, for which they develop the dynamic equations. Also they develop the dynamics of the neural controller in general terms and show the results in phase space when phase reset is applied and when it is not. Two effects are verified: 1. The phase reset in a suitable magnitude can relocate the system within the attractor stability basin after it has been put under a disturbance, obtaining stability; and 2. It can be useful to diminish convergence times to stable state.

HÃ$\frac{1}{4}$lse et al. [18], aimed to show that minimum and robust control structures can be developed using algorithm ENS3 presented by them, of evolution of neuro-modules made up of neurons with feedback connections , i. e. recurrent neural networks. To such networks an evolutionary algorithm is applied, where evolutionary operators are applied in each iteration: reproduction, variation (a type of mutation), evaluation and selection. In that process changes in the number of hidden neurons, their number of connections and the weights of such connections, are made. Once reached a satisfactory performance, there are reduced the number of neuro-modules elements, introducing costs by neuron and by connection, to urge the minimalism of controllers. The application of the complete procedure is shown in three examples: the expansion of a mobile robot for obstacle evasion to luminance tropism, the morphologic and control evolution for a biped walker robot with minimum actuation, and the co-evolution of neuro-controllers for a ring of gravitational impulsion with five actuator arms. An additional initial example is shown which corresponds to robot evolution for obstacles evasion and serves as guide for the controller development methodology according to the proposed algorithm.

Jha et al. [19] made the controller design for stairs ascension of a biped robot modeled in 2D. They use two fuzzy logic controllers to conserve robot stability according to dinamic stability margin criterion based on zero moment point (ZMP). The first controller is in charge to maintain stability during single support phase and the second to maintain it in double support phase. The controllers are, in a first approach, manually designed, and later the obtained controllers are optimized using genetic algorithms (GA), and in a third approach they are designed entirely by the AG. The low run times of the optimization suggest their possible application in real time, but it is necessary to implement ZMP control compensation when the controller does not manage to conserve the stability.

The approach taken by Kurz and Stergiou [25] to biped walking emphasizes the chaotic properties, already stood out by other authors, proposing them like a beneficial characteristic for control. They indicate that the capacity of a chaotic system to present equilibria with different periodic characteristics, added to the possibility of changing from one to another applying a small located actuation, facilitates the system robust control in presence of disturbances. They implemented a neural network whose inputs correspond to positions and speeds of initial states for the 8 previous periods of gait and its output is the hip actuation applied as control. The concluded that biological control of human walk can respond to similar phenomena, considering a more complex hierarchic structure in the neural network control.

Sabourin and Bruneau [37] applied a control based on trajectory tracking generated by a CMAC neural network to control the biped walking. Initially they calculate a set of rules or phases, of active and passive type, which generate partial actuation in different joints according to the geometric configuration in which is the biped walker robot. Later they trained a CMAC network so that it generalizes the trajectories needed to generate the walk, particularly for swing leg. The knee of supporting leg is blocked and its angle is determined by a high level control that, receiving a speed as parameter, modifies the angle to obtain a gait with the desired speed. The tracking of trajectories generated by high level control (support leg) and CMAC network (swing leg) is implemented with a PI control. It is shown the method robustness putting it under disturbances in floor surface, applying loads and sliding.

Computational intelligence methods mainly used are:

- Recurrent Neural Networks: For dynamic control and, in a novel approach, for a dynamical systems perspective of planning.
- Fuzzy Systems: For dynamic control.
- Hybrid Methods: Control, occasional correction of movements and learning.
- Evolutionary Algorithms: For optimization of planning and control schemes and as an alternative to learning methods.

2.5.4 Static Walking Control

The static walking refers to the achievement of locomotion by conserving static stability along all the path. It causes that a characteristic of locomotion patterns derived from this method is the very low velocities required to make negligible the inertial forces that can be generated, in such a way that they can be classified as quasi-static walking methods. This methods have been displaced by most elaborate dynamic methods, but they still have a research value in studying the stationary phases and posture of bipeds and some specific movement patterns.

Kun and Miller [24] studied again the biped walking problem, but now in static balance. Unlike [23], in this work they aimed for biped locomotion with quasi-static motion and, therefore, the stability criterion required that the projection of the CoM was at any moment within the supporting polygon, thus being able indefinitely to remain in any position. The trajectory generation, like in their previous work, is made with CMAC neural networks, starting off of pre-calculated but adaptable trajectories. In this case a network for foot elevation, one for forward movement and another one for the lateral motion are used. As the kinematics model used is an approximation, other two CMAC networks are used to make corrections to gait patterns such that they compensate model inconsistencies. The generator accompanied with a low level control that is in charge to map the generated posture to joints space, to make the trajectory tracking, to hold the perpendicular position to the ground in the double support phase and to make a lateral and frontal reactive control. A satisfactory gait in a 10-DOF robot with a speed of 2,2 passages per minute was obtained. Is remarkable the similarity of this method with that used in [23] despite of the dynamic orientation of the latter.

Miyakoshi et al. [28] studied the problem of open loop stabilization in periodic movements using neural network oscillators. They showed the satisfactory application to a juggling problem, and also to the biped walking with static balance (stepping). In this last one, it is used a pattern generator with an oscillator for the frontal plane and two coupled oscillators for the sagittal plane, as well as a PD position control with inhibition from the sagittal plane oscillators. Also they applied the method to biped walking in dynamic balance but obtain only some steps, since the method becomes unstable. They propose to deepen the study of the open loop control of the dynamic biped walking. This work exposes a divergence from the mainstream of walking control methods in that it emphasizes the role of open loop control and argues to its favor.

Chaisukkosol and Chongstitvatana [9] used a genetic algorithm (GA) to generate a walking sequence in static balance for a biped robot with counterbalance. The movement of the robot is made at a speed of 1 step every 40 seconds, reason why the system must be in static balance. The representation of each individual of the genetic algorithm contains the length of the chain and the series of angular displacements for the different actuators. The evolution was made sequentially for each one of the 6 phases that the authors define, initiating the evolution from a phase where the best individual of the previous phase evolution ended. A global fitness function is used which evaluates stability added with a particular function that evaluates the performance for the objective of each phase. The result of each phase is verified with the execution in a real model. This particular application is illustrative of the work enclosed by this global walking control methodology, but provides little general contribution to the field.

Wolff and Nordin [41] presented the evolution with evolutionary strategies (ES) of a control sequence to obtain biped walking with static stability for a robot. The evaluation of the fitness function is made directly in the physical system using a camera to measure direction and an infrared sensor to measure range. The evolution is made using a steady state algorithm which eliminates some of the individuals of an iteration, and manually makes the search in a zone defined by the Euclidean distance to a certain gene in the generated sequence. It is also shown that the evolved controllers performed better than the designed ones. The main contribution of this work is the physical fitness function evaluation which is seldom found in works on the field.

In another work Wolff and Nordin [42] they tried to generate a pseudo-dynamic walking, evolving points that are statically stable in the double support phase and interpolating in the balance phase. For it they make a evolution using evolutionary programming (EP) and representing the controller as a series of instructions with basic arithmetic operators and trigonometric operators, all of them represented like chains of integer numbers. The sequences thus evolved later are transfered to the robot, which can make fine adjustments or recovery of failures in a evolution second phase. The robotic platform used was the same one used in [41]. This last method is a hybrid method between dynamic and static walking control, in which an notable online failure recovery strategy is implemented, but the general characteristics are still in disadvantage with dynamic walking control methods.

The two mayor computational intelligence methods used in static walking are:

- Neural Networks: To compensate and map trajectories.
- Genetic Algorithms: To optimize motion sequences.

The owls they were watching.
The owls didn't care.
Then the owls came a knocking.
The Mars Volta (2001-)

3 Neural Fields as Control and Planning Systems

3.1 Introduction

In this chapter we aim to propose a control planning system or architechture based on neural fields which is suitable to control a relatively complex system. We test it over the stability problem on a typical inverted-pendulum and compare it against a more traditional recurrent neural network controller. First, we present the neural fields model, some variations of it which will be useful for its evolution (detailed in next chapters) and some of the properties that arise from it. Next, we study its applications to control and compare it with the recurrent neural network control scheme. We briefly compare its properties with traditional control schemes, and finally we test it with the inverted-pendulum problem.

3.2 Neural Fields for Control and Planning

For the purpose of control and planning we need some particular requirements on the neural fields.

The first one is to have a preprocessing over the input obtained from the sensors, so that there is a closed loop where the representation of inputs has an appropriate form. This mechanism alone (a particular form for the inputs) has shown to be enough for the robot ARNOLD to navigate in the plane with obstacles [5].

The second one is to be able to modify the connection kernel so that it can be suitable to our control problem. In order to do that, we will consider that the connection kernel $w(y)$ is a symmetric function (i.e. $w(y) = w(-y)$), that also is a 2-power Lebesgue integrable function so that it also belong to L^2. It can be shown that, whit that definition, a sum of an arbitrary number of kernel functions will also be a kernel function. This way, we have a inner-product defined by the Lebesgue measure:

$$\langle f, g \rangle_{L^2} = \int_{\mathbb{R}} f \cdot g d\mu \tag{3.1}$$

The defined space, whit its measure, conforms a Hilbert space, and therefore is complete and metrizable. It also gives a notion of sum, and scalar product:

$$(f+g)(x) = f(x) + g(x) \tag{3.2}$$

$$(\lambda f)(x) = \lambda f(x) \tag{3.3}$$

Those properties will be used shortly when arises the problem kernel evolution.

The third one consists of its suitability to simulation. This is not an inherent restriction for it to be physically (or biologically) plausible, but to be implementable on a computer. We will take a discrete form of the equation 2.2:

$$\tau \dot{u}_i = -u_i + \sum_{x_j \in B_p(x_i)} w\left(x_i, x_j\right) f\left(u_j\right) + S(x_i, t) \tag{3.4}$$

Where we replace the integral for a sum over the point included inside a finite neighborhood (ball) around x_i with radius p. The time is considered continuous, and the computation of the dynamical system behavior is evaluated with a Runge-Kutta method. We denote $u_i = u(x_i, t)$. It should be noted that the previous equation can be applied to the n-dimensional case without modification.

3.3 Control Architecture

The control architecture built based on the neural fields has three basic elements.

The first one, is a sensor, which reads the states from the plant and also their derivatives (computed from the dynamical equation of the plant). In particular, the sensor used for the neural field controller is based on the angular acceleration of the pendulum pole, loosely resembling the vestibular system on the inner ear.

The second one is the input layer, which consists of a simple neural field without natural dynamics, where the spatial codification of the sensed values is made. For the problem at hand, we use a finite one-dimensional neural field, where a sensed input with value zero maps to the center point of the field.

The third one is the processing layer, which has a more typical neural field which has inner dynamics given by the eq. 3.4, where the fields taken into the sum are the input neural field, and the processing neural field. This way, besides its natural dynamics, the processing layer receives the inputs from the input field filtered by the kernel operator. The kernel operator used is a Wizard Hat Function with the expression:

$$w(x_i, x_j) = ke^{-(x_i - x_j)^2/\delta^2} - H_0 \tag{3.5}$$

The additional term on the eq. 3.4 $S(x_i, t)$ is used only as the uniform and static resting potential, that is $S(x_i, t) = -r_p$. The firing rate function $f(u_i)$ is simulated as a simple Heaviside function.

The figure 3.3 shows the input and output layers (in the 2-dimensional case for generality) and the participation on the potential of a single element in the processing

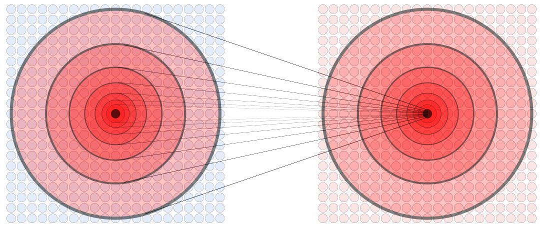

Figure 3.1: Neural fields for stability control

layer from the elements in the same layer and in the input layer.

3.4 Evolution of Neural Fields

3.4.1 Neural Field Controller Architecture

The architecture used for the neural field controller uses a structure similar to that of multilayer perceptrons, i.e. an input layer, a hidden layer and an output layer. The hidden layer has the properties so far presented in the neural field model. The input and output layers are also modelled as a population of neurons but without inner dynamics. Nonetheless, it is used a kernel for the connection from the input layer to the hidden layer, as well for the connection from the hidden layer to the output layer. The input layer is used as a buffer where sensory inputs are placed before they are processed by the hidden layer. The output layer is used so that it can be applied some form of post-processing to the output of the hidden layer without changing the inner dynamics of the neural field.

3.4.2 Evolutionary Algorithm Structure and Parameters

For the evolution process it is used a simple evolutionary algorithm as shown in the preliminaries, with random elimination of individuals inversely proportional with its fitness.

The evolution parameters are the connection kernels between the input layer and the hidden layer, and between the hidden layer and the output layer. The recurrent connections of the hidden layer with itself are left fixed, in the form of a wizard hat function.

The connection kernels are considered isotropic and homogeneous along the field, so that they can be described as symmetric one-dimensional arrays of values.

3.4.3 Genotypic Representation and Evolution Operators

Each connection kernel can be represented as an array of N values from $w(0)$ to $w(p)$ with homogeneous spacing, using its symmetry. This way, for an equal boundary radius for all the kernels, and a 3-layered architecture, there are $3N$ real values in the genotype. As can be seen, the number of evolution parameters does not have a direct relation with the simulation size of the neural fields (the number of discrete points used), in contradistinction with recurrent neural networks, where the number of parameters depends on the number n of neurons with a polynomial order $O(n^2)$.

3.4.4 Fitness Functions

The fitness functions were selected in such a way that the stability controller only has the goal to reduce inclination, while the positioning controller has to take into account both inclination and position. The fitness functions were tuned experimentally to attain a convergence velocity suitable for the experiment. This has in mind a notion of sequential evolution of, first, the capability to attain equilibrium, and later, the capability to perturb the equilibrium controller in such a way that a planned objective can be reached.

The fitness function for the stability controller is:

$$F_1(\theta) = 100 - \frac{100\theta^4}{\theta^4_{max} T_{total}} \tag{3.6}$$

And for the positioning controller is:

$$F_1(\theta, e_x) = 100 - \frac{100(\theta^4 + e_x^4)}{(\theta^4_{max} + e^4_{x,max}) T_{total}} \tag{3.7}$$

3.5 Experimental Framework

The model used consists of an approach to biped walking based on a inverted pendulum (car-and-pole) system in which the pendulum equilibrium is looked for. Nonetheless, supposing that the pendulum mass represents the body center of mass, it is proposed that is reasonable to expect a system with its sole function being to stabilize the body. This way, the navigation system has as purpose to carefully perturb the first controller in such a way that the stabilizing controller moves the car to the desired position.

3.5.1 Dynamic Model

The dynamic model used, in mathematical terms, is expressed in the two equations:

$$\ddot{x} = \frac{F + ml\dot{\theta}^2 \sin\theta - mg\cos\theta\sin\theta}{M + m\sin^2\theta} \tag{3.8}$$

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - F\cos\theta - ml\dot{\theta}^2\sin\theta\cos\theta}{l(M + m\sin^2\theta)} + \frac{\tau}{ml^2} \tag{3.9}$$

This model consists of four state variables and a high non-linearity as it departs from equilibrium points. It is worth noting that the wanted equilibrium point is in fact unstable.

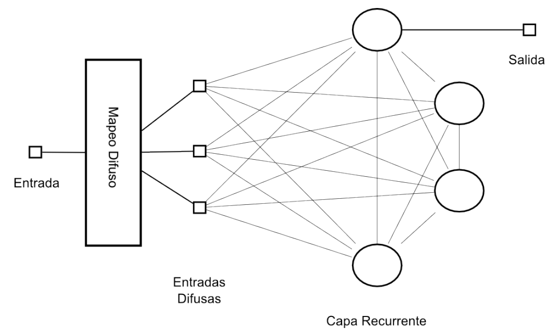

Figure 3.2: Neural net for stability control including fuzzy mapping

3.6 A RNN Approach for Comparison

The proposed architecture for the recurrent neural network controller has two expert recurrent networks, whose interaction will achieve positioning and equilibrium as well.

There has been applied a preprocessing stage previous to the input neurons, so that the actual values are not used and instead the inputs are mapped to 3 fuzzy sets. In this way, the stability controller only has 3 inputs, while the positioning controller has 6, corresponding to the same 3 inputs previously described and another 3 due to the fuzzy mapping of the error signal. All neurons are interconnected and the first one of them is selected as output without loss of generality. The neural network topology for the first controller (stability) is shown in the figure **??**.

3.6.1 Evolutionary Algorithm Structure for the RNN Controller

It is expected, based on the approach of artificial life to evolutionary robotics (Nolfi y Floreano), that the sequential and cooperative evolution of elements with biological similarity leads to an specialization in the process of stabilization and positioning (despite the antagonistic individual goals of each controller because of the interest of the positioning controller to maximize also the global performance).

As said, the two steps are executed sequentially, taking the best individual of the first step to collaborate with the individual evolved in the second step.

Aiming to obtain a fixed length representation and limit the problem dimensionality, it is used a model of order Q totally connected. Any network with an order equal or lesser and with total or partial connections can be represented by the proposed model, by the addition of activating/deactivating elements for neurons and connections. Therefore, individual are codified as:

- A bit sequence representing a serialization of an activation matrix A_a of dimension Q-by-Q which activates/deactivates a recurrent connection.
- A sequence of real numbers representing a serialization of matrices W_a and W_b, of dimension Q-by-Q and Q-by-$(m+1)$ respectively.

The C matrix is not evolved because it is chosen arbitrarily only one output (the first neuron).

The evolution operations used in both steps are:

- Parametric mutation of inputs: Gaussian modification of real codified matrix weights, which varies connection weights of inputs.
- Parametric mutation of recurrences: Gaussian modification of real codified matrix weights, which varies connection weights of recurrences.
- Selection: Calculates population fitness, selects with elitism and culling (5% of both) couples of parents for generating new offsprings, calculates the fitness function for both offsprings.

The fitness functions used are the same presented for the neural field controller.

3.7 Experimental Set-up

The model used consists of an approach to biped walking based on a inverted pendulum (car-and-pole) system in which the pendulum equilibrium is looked for. Nonetheless, supposing that the pendulum mass represents the body center of mass, it is proposed that is reasonable to expect a system with its sole function being to stabilize the body. This way, the navigation system has as purpose to carefully perturb the first controller in such a way that the stabilizing controller moves the car to the desired position. Here we are particularly interested only on the stability problem and controller.

Dynamic Model

The dynamic model used, in mathematical terms, is expressed in the two equations:

$$\ddot{x} = \frac{F + ml\dot{\theta}^2 \sin\theta - mg\cos\theta\sin\theta}{M + m\sin^2\theta} \tag{3.10}$$

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - F\cos\theta - ml\dot{\theta}^2\sin\theta\cos\theta}{l(M + m\sin^2\theta)} + \frac{\tau}{ml^2} \tag{3.11}$$

This model consists of four state variables and a high non-linearity as it departs from equilibrium points. It is worth noting that the wanted equilibrium point is in fact unstable.

The output from the stability controller maps to the lateral force F. The angular actuator with value τ is left to a value of zero, to allow the plant to behave according to its natural dynamics on the angular coordinate.

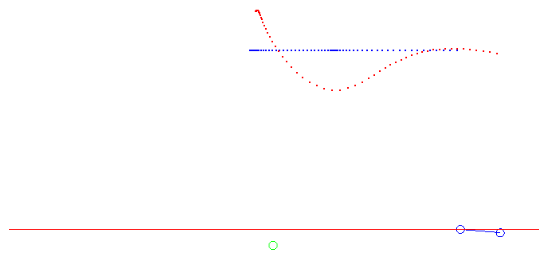

Figure 3.3: System dynamics with an untrained controller

3.8 Experimental Results

Experimentation Details

The sampling time used was 0.04s (for neural networks, neural fields, and visualization) and were performed 20s tests.

The differential equation system was solved by a numerical method, 4th Order Runge-Kutta. The iteration step selected was $h = 0.002s$ for each test.

Here are shown the results for the proposed neural field architecture without evolution and an appropriate selection of parameters (made taking in account the self-stability of the neural fields and the time constants of the plant), and the evolution of a recurrent neural network of a recurrent neural network controller.

Results

The first experiment tests the physical model using the recurrent neural network controller without evolution, to see the natural dynamics of the system when the controller is configured arbitrarily (in such a way that can be perceived the need of the evolutionary algorithm for the recurrent neural network controller). Results are shown in the figure 3.3. As can be seen, it is an unstable system in the origin. Red dots represent the pendulum position referenced to universal coordinates, and blue dots represent the base (car) position.

After the first step of the algorithm, and once done the stability controller evolution, it is shown the behavior withdrawing the positioning controller in the figure 3.4. The evolution was performed with a population of 50 individuals and 300 iterations.

On the other hand, when the initial angular perturbation is small, the neural field is able to control the stability without evolution. The simulation is shown in the figure 3.6.

The output from the processing field can be seen in the figure **??**.

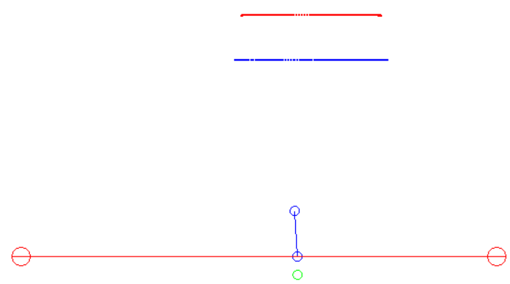

Figure 3.4: System dynamics with a RNN controller trained.

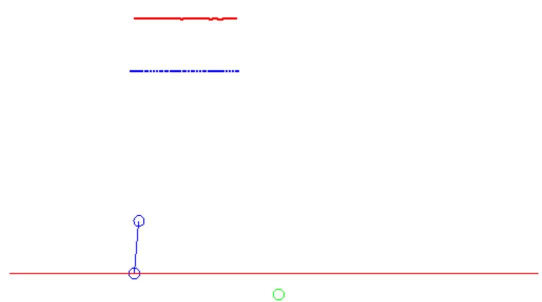

Figure 3.5: System dynamics with a non-trained Neural Field Controller.

3.9 Discussion

The results obtained from this chapter can be summarized in a short analysis.

While the recurrent neural network controller is expressive enough to solve the problem at hand, the number of parameter to configure (or in this case to evolve) is of a quadratic order in relation to the number of nodes (or neurons). This were not a particular problem for the evolutionary algorithm used, but limits its potential scalability. Furthermore, while it is expressive enough, does not show a particular suitability to the dynamic stability problem of the inverted pendulum and there are no reasons to expect something different for a more complex biped model.

On the other hand, the neural field controller is a bit more complex and its simulation more costly, but has some notable advantages. The first one is its ability to self-compensate or, equivalently, the stability of its natural dynamics, which is attained after the setup of few parameters. The second one is it suitability to the problem at hand, being able to solve it with a acceptable degree of performance for small perturbations. Although there was a need to parameter configuration, evolution was not required because the small number of parameters to setup: basically three parameters

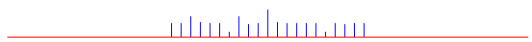

Figure 3.6: Processing Neural Field simulation.

of the kernel function and the resting potential of the field equation - a number of parameters of constant order in relation to the number of nodes (point potentials on the neural field).

It is left for next chapters the challenging but promising task of neural field evolution.

Exosqueletal junction at the railroad delayed.

The Mars Volta (2001-)

4 Adquisition of Goals by Evolution of Neural Fields

4.1 Introduction

4.2 Evolution of Neural Fields

4.3 Topological Structures for Multiple Goals

4.4 Co-evolution and Complexification

4.5 Acquisition of Objectives by Evolution

4.6 Experimental Framework

4.7 Experimental Results

4.8 Discussion

Will those shadows glare?
The Mars Volta (2001-)

5 Goal-Based Control and Planning in Biped Locomotion

5.1 Introduction

5.2 Computational Intelligence Methods in Biped Locomotion

5.3 Neural Fields for Control in Biped Locomotion

5.4 Neural Fields for Goal-Based Plannning in Biped Locomotion

5.5 Adquisition of New Goals by Evolution

5.6 Sustainable Learning by Evolutionary Landscape Adaptation

5.7 Experimental Framework

5.8 Experimental Results

5.9 Discussion

Falling Wormholes.
The Mars Volta (2001-)

6
Conclusions

6.1 Main Contributions

6.2 Future Work

Bibliography

[1] Asami, S. (1994). Robots in japan: present and future. *Robotics & Automation Magazine*, **1**(2), 22–222–26.

[2] Azevedo, C., Poignet, P., and Espiau, B. (2004). Artificial locomotion control: From human to robots. *Robotics and Autonomous Systems*, **47**(4), 203–223.

[3] Basdogan, C. and Amirouche, F. M. (1996). Nonlinear dynamics of human locomotion: from the perspective of dynamical systems theory. In *American Society of Mechanical Engineers, Petroleum Division (Publication) PD*, volume 77, pages 163–168. ASME.

[4] Benbrahim, H. and Franklin, J. (1997). Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, **22**(3-4), 283–302.

[5] Bergener, T., Bruckhoff, C., Dahm, P., Janßen, H., Joublin, F., Menzner, R., Steinhage, A., and Von Seelen, W. (1999). Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks*, **12**(7-8), 1087–1099.

[6] Cao, M. and Kawamura, A. (1998). A design scheme of neural oscillatory networks by hierarchical evolutionary calculation for generation of humanoid biped walking patterns. *Advanced Robotics*, **12**(7-8), 697–710.

[7] Capek, K. (1973). *R.U.R.* Pocket Books, New York.

[8] Capi, G., Kaneko, S., Mitobe, K., Barolli, L., and Nasu, Y. (2002). Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. *Rob Autom Syst*, **38**(2), 119–128.

[9] Chaisukkosol, C. and Chongstitvatana, P. (2001). Automatic synthesis of robot programs for a biped static walker by evolutionary computation. Second Asian Symposium on Industrial Automation and Robotics.

[10] Cheng, M.-Y. and Lin, C.-S. (2000). Dynamic biped robot locomotion on less structured surfaces. *Robotica*, **18**(2), 163–170.

[11] Dario, P., Guglielmelli, E., and Laschi, C. (2001). Humanoids and personal robots: Design and experiments. *Journal of Robotic Systems*, **18**(12), 673–690.

[12] Duysens, J., Van de Crommert, H., Smits-Engelsman, B., and Van der Helm, F. (2002). A walking robot called human: Lessons to be learned from neural control of locomotion. *Journal of Biomechanics*, **35**(4), 447–453.

[13] Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M. (1998). The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering*, **120**(2), 281–286.

[14] Garder, L. M. and Hovin, M. E. (2006). Robot gaits evolved by combining genetic algorithms and binary hill climbing. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1165–1170, New York, NY, USA. ACM Press.

[15] Hasegawa, Y., Arakawa, T., and Fukuda, T. (2000). Trajectory generation for biped locomotion robot. *Mechatronics*, **10**(1-2), 67–89.

[16] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

[17] Hermini, H., Rosa?rio, J., and Cassemiro, E. (2001). Proposal of modeling, simulation and implementation of robotics leg prosthesis. In *Annual Reports of the Research Reactor Institute, Kyoto University*, volume 2, pages 1415–1418.

[18] Huelse, M., Wischmann, S., and Pasemann, F. (2004). Structure and function of evolved neuro-controllers for autonomous robots. *Connection Science*, **16**(4), 249–266.

[19] Jha, R., Singh, B., and Pratihar, D. (2005). On-line stable gait generation of a two-legged robot using a genetic-fuzzy system. *Robotics and Autonomous Systems*, **53**(1), 15–35.

[20] Juang, J.-G. (2001). Intelligent trajectory control using recurrent averaging learning. *Applied Artificial Intelligence*, **15**(3), 277–296.

[21] Juang, J.-G. (2002). Intelligent locomotion control on sloping surfaces. *Information Sciences*, **147**(1-4), 229–243.

[22] Komatsu, T. and Usui, M. (2005). Dynamic walking and running of a bipedal robot using hybrid central pattern generator method. In *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, pages 987–992.

[23] Kun, A. and Miller, W. (1996). Adaptive dynamic balance of a biped robot using neural networks.

[24] Kun, A. and Miller, W. (1997). Adaptive static balance of a biped robot using neural networks.

[25] Kurz, M. and Stergiou, N. (2005). An artificial neural network that utilizes hip joint actuations to control bifurcations and chaos in a passive dynamic bipedal walking model. *Biological Cybernetics*, **93**(3), 213–221.

[26] Liu, Z., Li, C., and Xu, W. (2003). Hybrid control of biped robots in the double-support phase via h ? approach and fuzzy neural networks. *IEE Proceedings: Control Theory and Applications*, **150**(4), 347–354.

[27] Magdalena, L. and Monasterio-Huelin, F. (1997). A fuzzy logic controller with learning through the evolution of its knowledge base. *Int J Approximate Reasoning*, **16**(3-4 SPEC. ISS.), 335–358.

[28] Miyakoshi, S., Taga, G., and Kuniyoshi, Y. (2000). Stabilization of periodic motions.

[29] Miyashita, K., Ok, S., and Hase, K. (2003). Evolutionary generation of human-like bipedal locomotion. *Mechatronics*, **13**(8-9 SPEC.), 791–807.

[30] Nakada, K., Asai, T., and Amemiya, Y. (2003). An analog neural oscillator circuit for locomotion controller in quadruped walking robot. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pages 983–988.

[31] Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). A framework for learning biped locomotion with dynamical movement primitives. In *2004 4th IEEE-RAS International Conference on Humanoid Robots*, volume 2, pages 925–940.

[32] Park, J. (2003). Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy Sets Syst*, **134**(1), 189–203.

[33] Paul, C. (2003). Bilateral decoupling in the neural control of biped locomotion.

[34] Paul, C. (2004). Sensorimotor control of biped locomotion based on contact information.

[35] Paul, C. and Bongard, J. (2001). The road less travelled: Morphology in the optimization of biped robot locomotion.

[36] Reil, T. and Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Trans Evol Comput*, **6**(2), 159–168.

[37] Sabourin, C. and Bruneau, O. (2005). Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using cmac neural networks. *Rob Autom Syst*, **51**(2-3), 81–89.

[38] Taga, G., Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol Cybern*, **65**(3), 147–159.

[39] Vaughan, C. (2003). Theories of bipedal walking: An odyssey. *Journal of Biomechanics*, **36**(4), 513–523.

[40] Venkataraman, S. (1997). A simple legged locomotion gait model. *Rob Autom Syst*, **22**(1), 75–85.

[41] Wolff, K. and Nordin, P. (2001). Evolution of efficient gait with autonomous biped robot using visual feedback.

[42] Wolff, K. and Nordin, P. (2003). Learning biped locomotion from first principles on a simulated humanoid robot using linear genetic programming.

[43] Woo, S.-Y., Abramowitch, S., Kilger, R., and Liang, R. (2006). Biomechanics of knee ligaments: Injury, healing, and repair. *Journal of Biomechanics*, **39**(1), 1–20.

[44] Wu, Q., Sepehri, N., and He, S. (2002a). Neural-based control and stability analysis of a class of nonlinear systems: Base-excited inverted pendulums. *Journal of Intelligent and Fuzzy Systems*, **12**(2), 119–131.

[45] Wu, Q., Sepehri, N., and He, S. (2002b). Neural inverse modeling and control of a base-excited inverted pendulum. *Engineering Applications of Artificial Intelligence*, **15**(3-4), 261–272.

[46] Yamasaki, F., Endo, K., Asada, M., and Kitano, H. (2003a). Control design principle of a low-cost humanoid system using a genetic algorithm. *Adv Rob*, **17**(8), 779–790.

[47] Yamasaki, T., Nomura, T., and Sato, S. (2003b). Possible functional roles of phase resetting during walking. *Biological Cybernetics*, **88**(6), 468–496.

[48] Yanase, T. and Iba, H. (2006). Evolutionary motion design for humanoid robots. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1825–1832, New York, NY, USA. ACM Press.

[49] Yonemoto, K., Kato, I., and Shima, K. (1985). Technology forecast on industrial robots in japan. In -, volume 1, pages 51–58. Japan Industrial Robot Assoc.

[50] Zhou, C. (2002). Robot learning with ga-based fuzzy reinforcement learning agents. *Inf Sci*, **145**(1-2), 45–68.

[51] Zhou, C. and Meng, Q. (2003). Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy Sets and Systems*, **134**(1), 169–187.