

# Intelligent Approaches in Locomotion - A Review

Joe Wright · Ivan Jordanov

Received: 6 February 2013 / Accepted: 7 October 2014 / Published online: 26 October 2014  
© Springer Science+Business Media Dordrecht 2014

**Abstract** In this paper we review more than 140 publications and try to not only give a snap shot of the current state of the art research in the area, but also to critically analyse and compare different methodologies used in this research field. Among the investigated intelligent approaches for solving locomotion problems are oscillator based Central Pattern Generators, Neural Networks, Hidden Markov models, Rule Based and Fuzzy Logic systems, as well as Analytical concepts. We try to compare those methods based on the quality of the produced solutions in terms of time, stability, correctness and the expense and cost for achieving them. At the end of each section we list the advantages and disadvantages of the reviewed methods and scrutinize them considering the complexity of the approaches, their applicability to the investigated locomotion tasks and the constraints of the produced solutions. The reviewed publications examine a range of legged and non-legged systems, operating in simple and complex environments, with several different locomotion tasks.

**Keywords** Legged locomotion · Central pattern generator · Neural networks · Optimisation · Fuzzy logic · Hidden Markov models

**Mathematics Subject Classifications (2010)**  
82C32 · 60J20 · 93C42 · 68T40

## 1 Introduction

This paper presents a review of published locomotion control methods. The objective of the paper is to present a comparative overview of the field, so that the reader may make informed choices in investigating techniques suitable for their application. To accomplish this, each major technique identified in the literature is summarised, assessed and compared in the final discussion. Over 140 papers were reviewed, with comparison of techniques achieved by categorising them, establishing key properties and comparing in terms of scope of application, ease of parameterisation, and any desirable or undesirable characteristics. As many of the reviewed techniques require extensive parameter specification, this review also outlines key techniques of learning and optimisation as applied in the field.

The rest of the paper is organised as follows. In Section 2, analytical approaches are presented and subdivided into such that control bipeds (a major topic with demanding stability issues), and such that

---

J. Wright (✉) · I. Jordanov  
School of Computing, University of Portsmouth,  
Lion Terrace, Portsmouth, PO1 3HE, UK  
e-mail: badmintology@topracket.com

I. Jordanov  
e-mail: Ivan.Jordanov@port.ac.uk

control other types of systems. In Section 3, approaches inspired by biological solutions are presented and central pattern generators (CPGs), that create patterns based on oscillator primitives, are introduced. They allow more generalized ways of parameterisation of the movement patterns than the analytical approaches. Neural networks based solutions are discussed in Section 4. They offer an even more generalized approach than the CPGs and give more options for learning rules and optimisation. In Section 5, Hidden Markov Model (HMM) techniques are presented and their use for skill observation, classification and imitation is analysed. Transition table and fuzzy logic approaches are discussed as generalized state based methods in Section 6. Finally, a comparison and summary of the presented methods are given in Section 7. Numerous parameterisation techniques have been used in the literature, from simple manual specification, through to optimisation. A summary is given in each section, and an overall grouping of references by technique is given in Table 1.

Locomotion is the process of moving an organism or synthetic creature around an environment. Artificially producing locomotion is required in a range of disciplines including robot control [1–4], artificial limb control [5, 6], computer animation [7], and biological studies [8]. Locomotion may be needed for simulated models [1] or real world systems such as robots [9].

Generation of the movement patterns in locomotion is often a non-trivial task, for which a range of analytical and intelligent solutions have been developed [1, 3, 8]. The complexity of the problems ranges from producing straight line movement in a flat environment, to steerable navigation in an unpredictable environment, integrating perceiving sensory information and control. Controlled systems include bipeds and quadrupeds, as well as more unusual models such as fish, worms and others [10, 11]. A summary of references from this review, grouped by method, target system and type of data presented, is given in Table 2.

Beyond just one method of locomotion, some papers have investigated different gaits, ranging from walking to sprinting, along with smooth transitions between the gaits. As well as locomotion, some research has been conducted into other movement skills, e.g., jumping [12, 13]. We also review techniques that allow systems to imitate an observed movement pattern [114–118].

**Table 1** Reviewed references organised by control method, each of which are examined in separate sections of this review, and parameterisation technique

Method	Parameterisation
Analytical	Manual/simple: [1, 2, 9, 12, 55] Sequential quadratic programming: [56–60] Sequential surrogate optimisation: [61] Depth First search: [62] A* search: [63] Random search: [64] Gradient search: [64] Sequential search: [64] Genetic algorithm: [65–67]
CPG	Manual/simple: [3, 6, 10, 11, 46, 68–85] Exhaustive search: [84] Feedback learning: [86] Policy gradient search: [87] Actor/critic: [88] Sequential quadratic programming: [89] Hill climbing: [90] Genetic programming: [91] Genetic algorithm: [5, 8, 92–97] Hebbian learning: [98]
NN	Manual/simple: [99–101] Delta rule: [102] Genetic algorithm: [4, 65, 67, 103–106] Actor/critic: [107, 108] Self-scaling reinforcement: [109] Associative learning: [110, 111] Linear regression: [112] RPROP: [113]
HMM	Expectation maximisation: [114–116] Greedy policy: [117, 118]
Rule based	Manual/simple: [31, 119] Actor/critic: [120, 121] Genetic algorithm: [103, 121–124] Best-first search: [125]

The parameterisation techniques cover a range of approaches, from simple manual specification to evolutionary algorithms

Although some of the reviewed papers are easy to classify in a single category, there are also many that

**Table 2** Reviewed references organised by control method and target system

Method	Biped		Quadruped		Hexapod		Other	
	Sim	Hard	Sim	Hard	Sim	Hard	Sim	Hard
Analytical	[1, 14–32, 56–58, 65, 103]	[2, 9, 12, 13, 33–46, 59, 61, 66, 67]	[47–49, 62]	[50–53]	[54, 55, 63]		[64]	[60]
CPG	[5, 6, 68, 86, 92–94]	[3, 46, 69, 87–89, 95, 96]	[70–74, 98]	[75–79, 90, 97]	[80]			[8, 11, 81], [91] [10, 82–85, 126]
NN	[4, 65, 99, 102, 103, 112]	[67, 100, 109–111]						
HMM	[117, 118]	[114–116]						[104, 107, 108, 113] [101, 105]
Rule based	[31, 103, 119, 120, 122]	[121]			[123, 125]	[124]		

For each target system, the references are grouped into those that examine only simulated systems or theoretical work (Sim), and those that have tested the methods in real-world hardware (Hard)

span and overlap several categories. For example, specialized NN topologies blur the distinction between NNs and CPGs. Also, given their biological source, CPGs are often described as neural models, even if they do not resemble traditional NNs. Nevertheless, the aim of this review is to identify the core methods used in locomotion, and to discuss and provide comparative analysis of their usability, considering the application areas.

## 2 Analytical Approaches

The analytical approach is the oldest and most frequently met method in the reviewed papers. This approach relies on an understanding of the general form that locomotion in the target system should take. This may be based on observations of similar systems (such as human and animal locomotion) or from other analytical work. As a result, an algorithm is developed that explicitly states equations of motion (kinematic or kinetic) for the system.

### 2.1 Biped Trajectories

Bipedal movement starts from a simple premise - if the feet are placed in a forward moving pattern, and the rest of the body remains supported without falling to the ground, then the whole mass of the system will be moved forward continuously. Bipedal motion is therefore a combination of gait and whole (and especially upper) body stability. A typical procedure for constructing bipedal motion would be:

Plan a path to determine foot placements;

- Apply stability constraints to determine the Centre of Gravity (CoG) trajectory, based on a model of the weight distribution;
- Construct a plausible gait algorithm, addressing the double support (both feet on the ground) and single support (one foot off the ground) phases;
- Solve any remaining degrees of freedom (DoF) by any sensible manner. Methods may include copying human movement, simplifying movement, or even producing something that ‘looks right’.

The first and most common stability constraint is the Zero Moment Point (ZMP) [30]. It is calculated as the point under the foot where the ground reaction force will completely negate the effects of moments

and forces on the foot from the rest of the body (assuming sufficient friction). If the ZMP exists under the foot, then the system is stable, but if the calculated point is outside the foot, then the ZMP does not exist and the body (robot) will topple, rotating around an edge of the foot. The ZMP equations are used to ensure that the robot remains upright as the feet are moved. They are combined with a model of the mass distribution in order to determine a trajectory for the CoG. The model is often simplified to make deriving the equations simpler. A common model is the inverted pendulum model (IPM), which has a single point mass connected to the ground by a weightless rod [28]. In two dimensions this is given as:

$$\ddot{x}_{CoG} - \frac{(\ddot{z}_{CoG} + g)}{\alpha(z_{CoG} - z_{ZMP})}(x_{CoG} - x_{ZMP}) = 0 \quad (1)$$

where  $x_{CoG}$  and  $z_{CoG}$  are the  $x$  and  $z$  components of the Centre of Gravity respectively,  $x_{ZMP}$  and  $z_{ZMP}$  are the  $x$  and  $z$  components of the desired Zero Moment Point,  $\alpha = 1$  and  $g$  is the gravitational acceleration.

The gait algorithm is anything that raises a foot (now labelled as belonging to the swing leg) off the ground, moves it forwards, and places it back on the ground. Often, the mass models used assume that all of the mass is contained above the hip. Therefore, the CoG trajectory, as determined by the ZMP constraint, defines the hip angle trajectory. This is calculated to produce the CoG position relative to the hip position, which in turn is specified by the foot placements and leg joint angles. The foot trajectories are used to determine the leg joint angles and the overall motion is determined by the desired ZMP. It is located under the support foot in the single support phase and transitions to the other foot during the double support phase. Solving for the final trajectories is covered in [2].

Variations to the above procedure involve alterations to the stability constraint, alterations to the mass distribution model and alterations to the gait algorithm. Some of these variations are discussed below.

Equation 1 is a general form of the ZMP inverted pendulum constraint, but in many experiments it is simplified with the condition that there is no vertical movement of the CoG (i.e.,  $\ddot{z}_{CoG} = 0$ ). This is often referred to as the Three Dimensional Linear Inverted Pendulum Model (3D-LIPM). The simplicity of the 3D-LIPM algorithm made it popular in [2, 9, 24, 41, 43] (with an emphasis on controllability

including steering), [26] (with an observation system added), and [25] (using ZMP trajectory modification based on an analysis on the feasible ZMP region). In [1], a comparison was made between several different mass models. The authors found that all of the models could be written in the form of Eq. 1 but with  $\alpha$  varying depending on which mass model was used. Since different mass models will result in varying levels of accuracy to the true model, the identification of the constant  $\alpha$  allows the mass model to be fine tuned. By experimentally varying  $\alpha$ , the ZMP error can be reduced. As  $\alpha$  multiplies the height  $z_{CoG}$ , the model was called a Virtual Height Inverted Pendulum. A simple error minimisation procedure was used to find optimal values for  $\alpha$  for different step periods. The procedure involved incrementing or decrementing  $\alpha$  by a fixed amount, if the ZMP error was outside a threshold interval.

To further increase the accuracy, more complicated mass models can be used, at the expense of increased complexity of analysis. For example, multi-mass models were used in [31, 36]. In [36], three different bipedal control methods were evaluated and validated by comparing them to a reference multi-mass model with ZMP constraint. The compared models used polynomial interpolation between start and end states, actuator driving in the double-support phase, and a combined approach with added toe support and shock absorption, respectively. The toe support phase is omitted in normal ZMP based methods because it is, by definition, a failure with the foot beginning to rotate about the front edge. The authors included it to allow real-time freedom of choice for placing the landing foot. They argued this would allow for better walking on uneven ground.

Often, the ZMP is required to be under the centre of the foot during the single support phase, transitioning to the other foot in the double support phase. However, this is not the only way to position it as the ZMP can exist anywhere in the support polygon. Given this wider range of possible ZMPs, a control law for the ZMP was developed in [35]. The motion was based on the inverted pendulum model but the ZMP and vertical reaction forces were assigned control laws that allowed the centre of mass (CoM) to closely track a reference trajectory, allowing for real-time control and smooth motion.

In an attempt to generate more natural looking and efficient gaits, an energy control method was

developed in [15, 19]. The mechanical energy of the system was controlled to track a reference energy trajectory, which accounted for energy losses during the interactions between the robot and the surface. Trajectories based on conserving potential energy were developed in [37], and ease of analysis was suggested as an advantage of this method. Energy efficiency can be increased by creating self-excited behaviour in the swing leg [44, 45].

The ZMP constraint was developed for simple environment interaction, where the ground is even and the contact is only between the feet and the environment. Solutions to deal with uneven ground include an active balancing system based on distributing appropriate antigravitational forces to the contact points [34], and a modified ZMP technique using a nearest possible point to the desired ZMP during times when it cannot be found under the support polygon [28].

Extensions to the ZMP have been presented to deal with cases where other parts of the system are in contact with the environment. These include the Generalized Zero Moment Point [22], and Contact Wrench Sum [16, 32, 33]. Both of these techniques involve the relationship between all of the contacts made with the environment (feet, hands, etc...) and the forces through the system, with the goal of preventing unwanted rotations. They allow locomotion on uneven floors and the use of supports, such as handrails.

The ZMP can only be used to classify a state as stable or unstable. More informative alternatives include the Foot Rotation Indicator (FRI) [127], used to determine the stability margin or degree of instability, and the Centroidal Moment Point (CMP) [128], which provides information on the whole body rotational. After analysing how the various stability constraints performed for a human gait, the authors of [128] recommended a modified FRI that was more sensitive, and that the CMP and ZMP should both be used, for human-like locomotion. A reason to improve upon ZMP control is illustrated in [40], where significant differences were found between human and ZMP gaits, such as CoG trajectory, free leg trajectory, and the position of the ZMP under the foot.

Inspired by passive downhill walking systems, an active system was controlled, using a virtual slope walking gait in [39]. The system used knee bending and straightening to develop a relationship between the legs on level ground, that was the same as a normal gait on a downwards slope. The physical design of

the robot helped with stability and the control mechanism was found to be able to produce fast gaits. Quick gaits were also generated using a polynomial trajectory approach, with algebraic optimisation based on ZMP evaluation, torque and velocity limits [21].

If the system is dynamically stable, or controlled in a way to easily guarantee stability, then stability constraints are not needed [56]. In [23], a robot was designed that did not have actuators at the ankle and was stable enough not to require the ZMP constraint. Furthermore, to easily enable a high speed gait, each leg contained a telescopic actuator which pumped the swing leg.

The authors of [20] investigated initiating movement, stopping, and standing, to cope with real world state changes. Other transitions (e.g., for dealing with slopes) were examined in [38], where the algorithm had to control the walking on a flat ground, on a slope, and the transition between the two.

To help with walk stability, a method to control foot landing force was presented in [24]. It consisted of a system that measured impact force and outputted a foot height in order to absorb the collision. Full jumping and landing control was developed in [12] where, because of the point model used, ZMP or similar constraints were not possible due to the lack of a support polygon. Convergence to a known stable state guaranteed stability in the different phases of the motion. Another control algorithm for point feet was given in [58].

To avoid the over-constrained nature of ZMP type criteria, a different analytical framework was used in [57–59]. The analysis was based on transforming the problem into a task of driving a set of dynamic outputs to zero. Zero dynamics equations were produced for the swing and impact phases. The two phases were combined into a complete model that was called the hybrid zero dynamics (HZD). A suitable cost function of the HZD was defined and used to drive the motion into the swing and impact phases. Additional constraints were also imposed, such as minimal ground reaction force, foot landing only at the end of the swing, plus various boundary condition constraints. Given the cost function and constraints, a sparse sequential quadratic programming optimisation algorithm was used to determine the parameters of the model. Using these techniques, asymptotically stable walking (returns back to the gait after being perturbed) was produced.

Instead of synthesising the DoF trajectories, an alternative is to track supplied reference trajectories. In [27], a trajectory control law was developed to be resilient to perturbation. The authors identified a lack of resilience as a problem of previous tracking methods. Using human motion capture data was investigated in [42], and methods for interpreting the data automatically, for use by a humanoid robot, were developed.

Analytical approaches have occasionally been compared to other control methods. A ZMP inverted pendulum method (IMP) was used in [46] but, although successful, the approach produced slower walking speeds than an oscillator based control system. Instead of an inverted pendulum model, a static balance equation was used for the hip angle in [103]. This was implemented for a ditch crossing walk, and the foot placements were determined as a function of the distance from the ditch. The dynamic stability and energy efficiency of the analytical approach were compared to an optimized neural network, and fuzzy logic approaches. It was found to be the least stable and least efficient of the three approaches.

### 2.1.1 Trajectories for Non-bipeds

Bipedal movement is of significant interest because it allows control of human like systems, as well as posing challenging stability problems. In conducting this review, more papers were found examining biped gaits than any other. However, in this section we show that several different systems have also been examined. Most of the other systems examined are legged, but research can be found for specific control problems such as hand grasping [60].

With extra legs, stability becomes less of a problem, and the focus then shifts to developing gait algorithms – methods of moving the feet in order to move the system in a particular direction. Different types of gaits have been developed, to be used at different speeds, such as ambling, trotting, bounding and galloping [47–50]. In order to improve manoeuvrability, forward and crab (perpendicular) gaits for flat and sloped terrain were developed in [54]. Other research has looked at fine tuning the gaits, including comparing different modelling assumptions on final real-world accuracy [50], and increasing efficiency by utilising the natural dynamics of the system [14].

Some stability conditions have been considered in quadrupeds. For example, a set of reactive behaviours were developed in [52] in order to improve stability. In [55], an analysis of a spring loaded variation of the inverted pendulum model (also see [29]) was used to compare sensory feedback strategies for stabilising hexapod motion (as well as providing an analytical framework that could be applied to other systems). In order to be able to function in an uneven natural environment, and to support high speeds, a gait parameter and active-compliance parameter algorithms were developed in [51]. Gait adaptation was used to adjust to changes in the environment and provide a maximum stability margin, and compliance was used to deal with impulsive perturbations. The greater stability itself enabled faster movement across the uneven terrain.

In [13], the authors added two legs to a bipedal robot, in such a way that they were rigidly connected to the first to. This provided stability while only requiring a two-legged control algorithm. As well as creating a stable robot, the authors increased mechanical efficiency by introducing passive joint compliance. Altering compliance was done using pneumatic joints and, through exhaustive search, optimal compliance was found for walking, running and jumping gaits.

Control algorithms have been developed for unipeds. For example, in [64] a pogo stick style (a body plus one springy leg) robot was controlled with an established algorithm, but with the addition of variable correction term to minimize velocity error. The research in [53] examined the possibility of using uniped algorithms to control multi-legged systems (thus raising the possibility of utilising previous research in one domain across several). Firstly, they developed single legged gaits, which are gaits for multi-legged systems where only one foot is in contact with the ground at any one time. Following this they developed the concept of a virtual single leg that modelled the situation when two legs were in contact with the ground, this way expanding the number of possible gaits.

### 2.2 Optimisation

The specification of gait form and the use of conservative stability constraints usually limit the optimisation scope. However, there are some parameters available

to adjust, e.g., gait parameters including stride length and cycle frequency. These parameters can therefore be candidates for optimisation, according to some desired goal.

Such optimisation goals have included minimisation of energy expenditure [17, 58, 59, 65–67], minimisation of actuating effort [56, 57], improved form through minimisation of ZMP error [1], stability [58, 59], speed expressed as maximum distance over a set time frame [61], minimisation of speed error [64]. For those systems, which are lightly constrained in the analysis, constraints are imposed during optimisation to ensure walking gaits are formed [57]. Energy and effort goals are important due to difficulties in realising robots in the real world. Reducing the energy requirements makes robots more practical, and form and speed goals address their functionality.

Minimising actuating effort, defined as the integral square of actuator torques, instead of simple energy minimisation, was done for two reasons in [56]. Firstly, the quadratic nature of the fitness function favours smooth continuous actuator changes, which compares to actuator discontinuities that can result from energy minimisation. Secondly, an upright pattern was favoured, as it required only small torques to maintain.

The parameters for optimisation can be arbitrary constants, or they may have a direct interpretation. To have direct interpretation, the equations have to explicitly use understandable variables. This may require extra analytical work or result in more constrained solutions. Arbitrary parameters were specified in [1, 17, 56–59, 61, 64]. In contrast, specific end and intermediate target points of the gait were specified by optimisation in [65–67].

Various techniques have been used for optimisation, including Genetic Algorithms (GA) in [65–67], sequential surrogate optimisation (which works on an approximation of the objective function in order to speed up the optimisation – useful for real world cases) [61], sequential quadratic programming [56–59], and simple error based feedback [1]. For the velocity error minimisation problem in [64], the complicated algorithms derived in analysis were better optimized with random search when compared to gradient and sequential searches, due to numerical stability issues. In [62] a Depth First Search algorithm was used to plan the movement of a hexapod's legs

in order to navigate along a path, and in [63] an A\* search algorithm was used for similar reasons.

### 2.3 Advantages and Disadvantages of the Analytical Method

#### Advantages

- If the analysis is suitably accurate, solutions can work correctly right away;
- In general, there is no need for a training phase and this means implementation in real hardware is easier than for methods that require learning or evolving;
- Analytical approaches have been investigated for a long time and there is a diverse set of solutions, covering many task conditions and requirements;
- Solutions are directly interpretable and expert knowledge is a key feature.

#### Disadvantages

- The analysis used to construct gait algorithms is generally very task specific. Because of this, locomotion is unlikely to succeed under conditions not assumed or accounted for during the analysis;
- Likewise, behaviour is generally task limited and there is unlikely to be any emergent properties, e.g., gait transitions;
- Usually the models used in the analysis are approximate. This gives the risk of inaccurate, unstable, and/or inefficient movement compared to the aims followed during analysis;
- In order to simplify, or even enable analysis, solutions are generally over constrained and therefore will rarely be optimal. Some more advanced approaches, e.g., those using the hybrid zero dynamics framework, can produce less restrictive constraints;
- The limited number of parameters reduces the effectiveness of optimisation. Desired goals, such as speed or efficiency, are hampered by the over constrained nature of the analytical approach. Often, the form cannot be significantly parameterized, only specific values can be changed (e.g., stride length);
- Over-constrained results can look unusual when compared to equivalent systems in nature.

### 3 Central Pattern Generators and Oscillators

In motor skills science there was a debate over whether locomotion is reflex based or is generated internally. Experiments in the second half of the 20th century demonstrated that internal generation had to be a significant part of locomotion [129]. This was proved by severing sensory neural pathways in animal subjects and observing that they could still perform locomotion. In more recent times, dissections have enabled a reverse engineering of the neural networks that control this innate locomotion. Those networks that have been discovered exist in spinal regions and have therefore been called ‘central pattern generators’, or CPGs [130–133].

A biological CPG can be defined as a neural network that produces rhythmic pattern outputs without the need for patterned input. However, a distinction should be made between CPG neural networks and more traditional NNs, which are discussed in Section 4. In reviewing the engineering use of CPGs, we have determined that it is better to see CPGs as systems of oscillators, rather than as neural networks. In biological systems, the primary unit of rhythm is built around a pair of inhibitory/excitatory neurons that produce oscillations. A detailed examination of a biological CPG from an engineering perspective was conducted in [134] but, for most applications, the neuron pair is approximated with a pair of differential equations.

The review of CPGs here focuses on application to robotic control and generalises the concept to oscillator models. For other perspectives, including historical and biological contexts, see the reviews in [135, 136]. Some approaches, that are referred to as CPGs, do not even explicitly use pairs of differential equations, but rather use oscillators with more transparent sinusoidal forms, as discussed in the next section.

#### 3.1 Oscillator Types

##### 3.1.1 Simple Sinusoidal Systems

These systems are based on simple sinusoidal oscillators. Coupling terms added to each oscillator equation allow phase relationships to be established between them. In [10], this form of oscillator acted as the basis of a CPG controlling a swimming robot. Oscillators with amplitude and angle offsets controlled each fin.

Setting different values for these two controls enabled forward swimming, backward swimming, spinning, turning while swimming, turning on the spot, swimming up or down and crawling gaits.

For some systems, their locomotion comes from whole body sinusoidal movement. In order to replicate movement seen in nature, simple sinusoidal oscillators were used to produce forward, circular, turning and sideways movement in an eel robot [83, 126]. Sinusoidal control of serpentine motion on a planar surface was investigated in [84]. Of interest was the mechanical efficiency of the gait for a range of speeds, and optimisation was used to find the best values for the constants in the algorithm, under three different friction models.

For other systems, such as bipeds, sinusoidal motion does not initially seem a good option. However, in [3], simple sinusoidal oscillators were combined to construct gaits for two humanoid robots, one of which was human sized. The coupling of different oscillators allowed more complicated trajectory shapes, suitable for bipedal locomotion.

Systems with several legs (four or more) can be stable enough to allow a leg to be moved without worrying about the system falling over. Without stability requirements, the control mechanism can be very simple. For example, simple sinusoidal oscillators were used in [81] to control an eight legged robot simulation. A simple oscillator system coupled with feedback, investigated in [46], was able to produce faster walking speeds than a ZMP IMP approach.

##### 3.1.2 Systems of Differential Equations

Analysis of biological CPGs has identified oscillators made from pairs of mutually inhibiting neurons [137]. These can be modelled or approximated by systems of differential equations. The solutions of these systems, and variations on this theme, range from simple sinusoidal forms to more complex shapes.

Matsuoka oscillators are commonly used for CPGs [5, 68, 69, 72, 75, 79, 88, 92–95, 138], and one reason for their popularity is that their dynamics, including limit cycle behaviour, are well known [138]. Matsuoka oscillators are capable of different gaits, such as the walk, trot and pace quadruped gaits, by specifying different phase relationships [75]. They are also capable of smooth gait transitions, which were dramatically demonstrated in [72] where a 2D biped switched to



climb on all fours up the slope. Using sensory information, Matsuoka CPGs are able to control walking not just on a flat surface but on an irregular [75] or slippery surfaces [79].

Van der Pol oscillators (VDP) are also popular for CPGs [11, 73, 78, 89]. The benefits of these oscillators are that they have stable limit cycles and relatively interpretable coefficients. Frequency, amplitude and shape coefficients can be identified, although they are not completely independent. Different gaits are possible with VDP oscillators, including walk, trot, pace and bound quadruped gaits [73], walking and forward jumping biped gaits [78]. In both situations, the ability to transition between gaits was developed through simple changes to the parameters of the oscillators. Rayleigh oscillators, which are similar to VDP oscillators, were used in [6] to construct a CPG for an active prosthetic leg. Reduced sensory requirements were highlighted as an advantage over conventional prosthetic systems.

Similar to the Matsuoka oscillators in form, Amari-Hopfield oscillators are also used in CPGs [71, 87]. In [87] they generated biped locomotion that was resilient to perturbations as a result of the limit cycle properties of the oscillator. In [71], they were able to produce walk, trot and gallop quadruped gaits, with prompt gait transitions.

Hopf oscillators were used in [74, 90, 98] for quadruped locomotion. In [74], they were chosen for their stable limit cycle (even when the frequency changes), and for their independence of cycle shape and frequency. In [86], a series of Hopf oscillators was investigated, that included feedback terms allowing learning of an input trajectory, with each sub-oscillator matching a partial of the input. A CPG using these combined oscillators was trained with a reference bipedal locomotion pattern, thus converting a reference trajectory into a system with limit cycle properties, so that it became resilient to perturbation.

Spiking Integrate and Fire with Adaption neurons were used in [96] for biped locomotion. The CPG had a hierarchical structure of hip timing, knee timing, and finally output patterning. The core oscillator consisted of a neuron pair that controlled the hip timing, which in turn coupled to a pair for the knee timing, therefore establishing the gait characteristics. Finally, the output of the timing neurons was sent to sets of parallel motor neurons that had independent configurable firing characteristics. They therefore allowed production

of complex patterns that were driven by the periodicity of the timing neurons. This three sub-system design allowed separate configuration of walking frequency, gait and joint angle profile, by adjusting the parameters of the hip oscillator, the hip-knee coupling, and the motor neurons respectively.

Generalized CPGs were developed in [139] using piecewise affine systems. Analysis was given that allowed amplitude and frequency to be independently specified. The paper cited flexibility, and linear relationships between input frequency and amplitude controls and the output frequency and amplitude, as benefits of this approach.

Another type of biologically inspired oscillators is the Ellias shunting oscillator, which was used in [80, 140].

### 3.2 Parameter Selection

In general, the oscillator constants and coupling weights of the CPG need specifying. At the simplest level, manual specification can be used, as discussed in [3, 6, 10, 11, 68–70, 73, 74, 76, 78, 80, 82, 141]. Although this approach is unlikely to be optimal for any particular criterion, it may be sufficient to give good results. It allows solutions to be determined with few trials and is therefore appropriate for implementing directly in hardware. By analysing gait characteristics common to several animals of a type similar to the target system, variables such as phase differences can be determined. For oscillator types where there is good parameter separation, frequency and amplitude components can again be specified by observing or making logical arguments and conclusions.

Parameter optimisation criteria include minimising energy expenditure [84], maximising speed [87, 90, 97], matching a control speed [98], reference tracking [86, 89, 95, 96], maximising stability [5, 88, 92], and producing walking in an unconstrained system [93, 94].

The implemented optimisation techniques vary from exhaustive search [84], to hill climbing [90], policy gradient search [87], actor/critic [88], sequential quadratic programming [89], Hebbian style learning rules [98], integrated oscillator learning terms [86], and GAs [5, 8, 92–97]. The exhaustive search, hill climbing and sequential quadratic programming offer straight forward approaches but are limited in their application. The learning rules/terms provide real time

error correction or reference tracking, and are therefore possibly appropriate for direct hardware implementation. The gradient and actor/critic searches are used to speed up optimisation, while the GA are used to fully explore and exploit the search space and possibly find global optimal solutions, but require more computation.

Multi staged optimisation has been used to simplify the complex process [95, 96]. For example, in [95] a GA was used to find parameters for the hip joints first, then all joints in the left leg, the whole lower body, the upper body, and then finally, the whole body (which had 271 parameters). The optimized fitness functions can include multiple components [92, 97]. In [92], stability, pose angle and step length were optimized in order to find good solutions for stability, form and speed.

Using outcome measures to produce walking in an unconstrained system can be difficult. The optimisation process needs to find a workable solution from a large solution space. Using travelled distance as a fitness function was successful in [93] when combined with analysing final height, to detect possible falling, and average step length. In [94], the authors initially failed to produce a walking gait with the outcome measures of travelled distance, frequency of foot strikes, and uprightness. They presented a theory that control systems in nature may have co-evolved with the structure of the physical system. With this in mind, they added support structures to the biped that allowed it to evolve effective gaits without the risk of falling down. Although not investigated, the authors suggested that the supports could then be removed and optimisation continued.

A somewhat different approach was demonstrated in [91], where a genetic programming algorithm was employed to construct a sinusoidal system, producing optimal side winding locomotion for a snake model. The function set of the GP algorithm consisted of trigonometric and basic arithmetic functions, and the fitness function was the distance travelled in a set time.

### 3.3 CPG Advantages and Disadvantages

Advantages:

- In general, the oscillator nature of CPGs produces limit cycle behaviour. This means that often, CPGs are resilient to perturbation;

- Another feature of the limit cycle properties is that there can be emergent behaviour, such as gait switching with smooth transitions;
- CPG oscillators have a clear parameter structure that is open to optimisation;
- The oscillator based approach is less constrained than analytical approaches and therefore potentially capable of better solutions when optimized.

Disadvantages:

- Parameterisation often requires calibrating, training or evolving;
- It is sometimes difficult to interpret the method of the control system, compared to those derived analytically;
- Although less constrained than analytical approaches, CPGs are more constrained than some other solutions. This implies they have less optimisation potential than some of the other methods;
- Due to the oscillator structure, CPGs are mostly limited to cyclic behaviour. Non cyclic behaviour would require some starting and stopping mechanisms.

## 4 Neural Networks

Although the CPGs discussed in Section 3 are often described as neural networks, they are in fact highly specialized and better interpreted as oscillator models. In this section, the use of conventional neural networks (NN) for locomotion is discussed, and the NN are grouped into feed-forward and recurrent networks. As typically used, NN process data inputs and are therefore suitable for state based approaches. They are also capable of self generated patterns when using recurrent structures.

### 4.1 Feed-Forward Networks

Feed-forward neural networks (FFNNs), in which each neuron has a summing junction and a transfer function, have a straight forward input-process-output architecture. They can be readily applied to state based motion generation, where current kinematic or kinetic sensory variables can be processed through the layers of the network to generate actuator values. These networks can be used in continuous or discrete time,

and can optionally include external sensory information in the input layer. The lack of feedback within the FFNNs prevents them from implicit trajectory generation over time, but by providing a time input, a FFNN is able to generate a trajectory that is not state based.

#### 4.1.1 Multi-Layer Perceptron

In a multi-layer perception (MLP), data is fed forward through layers of neurons. A typical arrangement is to have an input layer, a hidden layer and an output layer, although more hidden layers are possible. Each layer after the input layer can be fully connected to the previous layer. So, a neuron in the first hidden layer takes a weighted sum of the inputs, plus a weighted bias value, and processes that information through an activation/transfer function, which is usually a non-linear, sigmoid type. Similarly, output neurons process a weighted sum of the output of the hidden layer neurons, plus another weighted bias, to produce an output. MLPs are then trained and configured by adjusting their weights, so that the produced output is close to the target.

In [103] (also mentioned in Section 2), two MLPs were used to specify parameters of a bipedal ditch crossing gait. The gait was largely constructed analytically but actuator values for the swing leg were left to be controlled by the NNs. The NNs were trained with GA and the best solution was found to be more stable and efficient than one from a fully analytical approach, and slightly better than the result obtained with a fuzzy logic based method.

In [65, 67] NNs were trained with a GA derived data set that contained optimal parameter values (for energy minimisation), for an analytical based stair climbing solution. The set had values for different step lengths of the robot and stair heights in the environment. Once trained, the NN gave an effective method to interpolate between the GA derived data, in order to approximate optimal values for new step lengths and stair heights.

#### 4.1.2 Radial Basis Function Network

The activity of the neurons in a MLP is distributed across the layer with potentially several neurons firing (giving a high output) for a particular input. In contrast, the neurons in the hidden layer of a radial basis function network (RBFN) are local and respond

strongly only to inputs in a particular part of the input space. To do this, they have a reference vector called the weight, or centre vector, which is of the same dimension as the number of inputs. Their activation is typically a Gaussian function of the Euclidean distance between their weight vector and the input vector. Each vector, then responds strongly when the input is close to its weight vector and weakly when the input moves away from the weight vector. If the Gaussian is sufficiently flat, then the output can offer generalisation for inputs that fall between the weight vectors of the hidden neurons. The output neurons are a function of a linear combination of the outputs of the RBF neurons and their weighted connections.

In [107], an RBFN was used to control hexapod locomotion. The inputs to the system were the sensor states of the robot at discrete time intervals and the outputs were the actuator actions, as well as an evaluation output used for actor-critic temporal difference training. The network was self-organising and expandable, so that sensor vectors that were different to those experienced previously could be accommodated. Training was done in stages; firstly single leg control was developed, followed by leg coordination. The network was interpretable enough that it could be seeded with expert knowledge, and this was found to greatly speed up the subsequent training.

In [108] reinforcement learning for a RBFN network alone was compared to such with the addition of analytical processing. The analytical addition reduced the number of outputs the RBFN network needed by incorporating prior knowledge. The hybrid approach produced faster training and well behaved characteristics in comparison to reinforcement only approach.

#### 4.1.3 CMAC

A Cerebellar Model Articulation Controller or CMAC network is a type of associative memory network based on the cerebellum [142]. The continuous input space is divided into hyper-rectangles so that an input is located in one rectangle at any one time. Multiple layers are used with the placement of the rectangles slightly offset for each layer, so a rectangle in one layer will overlap several in the other layers. In this way one input is associated with multiple hyper-rectangles, one in each layer, but changes in the input will result in different changes in activation in each layer. Each hyper-rectangle in each layer

has a weighted connection to the output neurons. The output of each node is the weighted sum of the activated rectangles, and the weights are adjusted through training.

In [109], a set of CMACs were used to generate locomotion. At the core was a pattern generator formed by a CMAC that took time, step length and walking period as inputs, and generated six desired joint angles as outputs, and was trained using least mean squares supervised learning to match a basic joint trajectory profile over one walk period. The system also used CMACs as peripheral controllers that helped maintain body posture and body height. Additional reinforcement learning was used by the pattern generator and peripheral controllers, based on several constraints: energy, step, posture and a body height constraint. The last two constraints were themselves calculated by CMACs that were trained to assess the current posture and body height states, and to output an appropriate evaluation. The additional learning enabled the robot to refine its movement and learn new solutions. The modular nature of the solution allowed for smaller, easier to train neural networks.

In [110, 111] CMACs were used to learn walking patterns derived by an analytical approach. The CMACs successfully learned the movement patterns and showed resilience to perturbation (including on uneven or slippery floors), thereby translating a rigid analytical solution into an adaptable one.

#### 4.2 Recurrent Neural Networks

Recurrent neural networks (RNN) recycle information, via time delays, back into themselves, which makes them useful for processing inputs that evolve over time. Of particular interest to locomotion is that they can also exhibit limit cycle behaviour, and self generated patterns, and so are capable of producing periodic trajectories.

Upon this simple premise, more complicated networks can be developed to produce complex patterns and handle different types of input. With the RNNs outlined here there is a partnership between processing external sensor data and internally generating patterns. Inputs into the RNNs for locomotion generally consist of gait selection and sensory information. For non-recurrent networks, the inputs are vital to produce the trajectories but, for the RNN, the pattern is

produced internally and modified or selected by the inputs.

##### 4.2.1 Jordan and Elman Recurrent NN

A modified *Jordan* network was used in [102], trained using the generalized delta rule [143] to match reference trajectories, indexed by different inputs to the control nodes. The network is similar to a standard feed-forward network except that the outputs are fed back into a set of input neurons for the next iteration, but without self-feedback connections normally present in *Jordan* NNs, as these were found to hamper the ability of the system to learn periodic trajectories. Accurate fault tolerant trajectories were learned, and the system could interpolate between forms found in the training set, by varying the control inputs accordingly.

*Elman* NNs are similar to the *Jordan* architectures, except that instead of recycling information from the output layer, time delayed copies of the hidden layer are maintained and fed back into the hidden layer. The *Elman* network was used in [113] and the authors claimed it had increased power (capable of a wider range of solutions) over the *Jordan* network. The resilient backpropagation (RPROP) training algorithm was used and the system showed an ability to learn supervised trajectories and interpolate between them.

##### 4.2.2 Fully Connected Recurrent NN

In [4], a network of ten fully connected leaky-integrator neurons was used as a pattern generator, where six of them were also connected to the actuators to output the trajectories. The output functions of the neurons insured changing outputs over time, even if the network was initialized with zero outputs for all neurons. Successful bipedal walking patterns were evolved using a GA, using distanced travelled in a set time frame as the fitness function.

A similar network was used in [106] where it was evaluated for properties that may be useful in locomotion, by learning sinusoidal patterns. The parameters of the network were again evolved with a GA. The number of generations required was found to be largely independent of the NN sampling rate (how many iterations the network should perform per second). This implied that the network learned the

form of the patterns rather than individual sample values and simply scaled output frequency to the sampling rate.

The network was able to learn multiple sinusoidal patterns, selectable by an input level, but evolution took more generations as the number of patterns required increased. Learned patterns were resilient to noise at the inputs, and even if the input was disrupted by a large amount, the output would quickly return to normal when the input was corrected. It was found that a six neuron model was sufficient to store eight different sinusoidal patterns, as long as the form of those patterns was different.

A six neuron fully connected RNN was also investigated in [105], to control an agent moving to the location of a chemical marker in two dimensional space. The network was trained with GA but, although successful, a large population size was needed to avoid convergence to sub-optimal local solutions. A sparsely connected set of RNNs (one for each leg), consisting of five neurons each, was then applied to produce an insect gait. The approach was later shown to be successful in a real hexapod robot in [104].

#### 4.2.3 Reservoir

Reservoir networks are different from traditional RNNs. Normally, RNNs should be as small as possible to help training, but reservoirs use a big network with potentially much redundancy. This succeeds, however, because the reservoir, which is a large set of interconnected neurons, remains unchanged over training - only the output neurons are trained. The assumption is that the dynamics needed are already present when the network is formed (randomly), and all that is required is to correctly listen for them by adjusting the output weights. This gives reservoir NNs the advantage of simplified training (for example, linear regression techniques can be used), albeit at the expense of somewhat inefficient operation due to the larger number of neurons.

In [112], a reservoir NN was used to generate trajectories that matched references taken from human motion capture. The network was trained with ridge regression, and noise was added during the training phase in order to produce a more stable pattern generator. The NN produced accurate trajectories over long time periods and was resilient to perturbation.

The system was also trained to have controllable frequency and managed to specify patterns for 22 DoF.

#### 4.2.4 Specialized

Partially hand crafted, and partially trained NNs were developed in [99] for pattern matching real life gaits, and in [100] to control a biped that was resilient to sensor noise and self generated sensor movement patterns. An analytically derived NN was created in [101] with no training or learning, for a hand grasping movement.

### 4.3 NN Advantages and Disadvantages

Advantages:

- They are the least constrained of the approaches in this review. Generalisation abilities can be improved by optimising the network topology, at the potential expense of greater processing load and training complexity;
- NNs have shown an ability to cope with situations outside those found in training. This includes good limit cycle behaviour in RNNs (resilience to perturbation) and an ability to interpolate between training inputs. Examples have been given where trajectories derived from an analytical approach have been learned by a NN, with the output of the NN then resilient to changes in the environment that were not accounted for in the original analysis;
- The ability to produce interpolated output when controls are varied allows for novel gaits to be generated, that were not originally trained or optimized for;
- The NN architectures are convenient for training or optimisation and, in general, this is how their parameters are specified. Optimisation techniques such as GA can be easily applied for NN training;
- Some of the FFNN structures are interpretable enough that they can be modified or seeded with expert knowledge.

Disadvantages:

- Many iterations are generally needed for training (optimisation) and learning so using a real

world platform becomes difficult. This means that simulation is often required;

- Many of the structures, especially the RNNs, are difficult to interpret. For practical implementations this may not be a concern, but it does mean development often cannot be aided with expert knowledge;
- Theoretical interpretations of the resultant methods are limited.

## 5 Hidden Markov

Some research has been conducted into robotic learning by imitation, where the robot goal is to observe a motion pattern and then to reproduce it. Often, a human may be the source of the motion pattern to be copied and because the physical workings and capabilities of the source and target systems are different, simple copying will not work. Imitation therefore, becomes a process of observation and re-synthesis. Here, observing is estimating the underlying state variables of the source system where only the output is available, and Hidden Markov Models (HMM) have been used in robotics for this task [114–118].

When imitating, the system first observes a movement pattern and a recognition algorithm is then used to determine if the movement is already known, in which case the pattern is used to refine the stored one. If the pattern is not recognized, then it is added to the database as a new, learned pattern. The observed kinematic or kinetic values, as well as the synthesized values, are called ‘motion elements’ in the HMM papers reviewed here. One motion element represents the kinematic/kinetic values at one discrete moment of time. The hidden states of the HMM provide an abstraction of the movement patterns, which can be used to re-synthesise the motion in the target system.

The first part of imitation is recognition, where a recursive algorithm calculates the probability of observing a movement pattern, if the candidate HMM was used to generate it. If this probability exceeds a threshold value (which can be varied to control grouping of similar observations), then the observation is determined to have fit the stored model and is, therefore, recognized. This recognition algorithm was modified in [115] to cope with missing elements.

This could be used if observation was temporarily occluded.

If recognition does not occur, then the next stage of imitation is learning. A new HMM is generated from the observation sequence using an expectation-maximisation algorithm (EM), such as the Baum-Welch algorithm [144].

To synthesize, or produce the movement once stored, it is usually generated stochastically from the HMM. Because of its probabilistic nature, typically, the synthesis is repeated several times, with each sequence normalized in time. The sequences are then averaged to produce a final output. In [115], the *Viterbi* algorithm [145] was used instead, to generate a sequence that most closely matches the observation.

Finally, an error value is generated based on the difference between the synthesized and observed sequences. This error value can then be used in a learning rule to modify the matrices of the HMM and refine the stored pattern.

Although typically used for imitation tasks, HMM have also been employed to detect problematic states. For example, in [116], a Gaussian Mixture Model method was used to identify the current state and a HMM method was used to classify a series of states over time. Once a problem, such as falling, was detected, a reflex algorithm was initiated in which both methods were successful with one version of the HMM approach being excellent at fall prevention, or at least producing a controlled fall.

### 5.1 HMM Advantages and Disadvantages

Advantages:

- HMM can be used as a specialist technique for movement control. They are well suited to imitation when compared to other techniques;
- They provide a way of recording underlying trajectory data to a high precision;
- HMM can be employed to identify problematic states and therefore, to initiate saving reflex programmes;
- The database structure found in the methods using HMM allows the ability to learn multiple skills. However, this is more a property of how they have been used in the literature than an inherent

advantage. Other methods could also have a database of parameter sets that allow them to perform multiple skills.

Disadvantages:

- HMM do not provide emergent behaviour, such as the gait changing found in some CPG and NN approaches. Indeed, as used so far, HMM approaches are a trajectory playback system – therefore not inherently adaptable;
- Expert knowledge cannot be implemented directly into HMM;
- Hybrid designs would be needed to add enhanced properties such as resilience to perturbation.

## 6 Rule Based Systems

By classifying the current state, the next action can be determined through a table lookup or rule based system. The simplicity of these approaches enables clear interpretation, and therefore the use of expert knowledge or learning/optimising techniques to specify transition rules.

### 6.1 Transition Table

In [124], a transition table was evolved, in order to specify actions based on the current system state, for a hexapod. Body states were classified into an integer value and this was used to index a list of actions, used to move the legs. For a given transition table, generated using a GA, the hexapod was simulated starting with each possible initial configuration, and the fitness function was the proportion of those starting positions that led to a tripod gait. The best table evolved gave stable tripod gates for 98.4 % of the initial states, with the failures arising from initial symmetrical leg positions that could not lead to asymmetrical tripod gaits.

A transition table was also used in [125] to control a hexapod, but this time, the state value was calculated as a binary string where each bit represented either supporting (down) or not supporting (up) leg. With one leg being raised or lowered at a time, the goal was to move through a terrain that had specified

bad patches (pre-known to the algorithm). Using a graph search technique, locomotion was successfully generated and exhibited different gates.

### 6.2 Fuzzy Logic Systems

A zero moment point controlled biped was developed in [31], where in order to improve the gait, a fuzzy logic controller was used to vary the ZMP position in the support rectangle of the foot. This was done to match the shifting in real human gaits from the rear to the front of the foot. After simulations, this technique resulted in a more desirable gait with less trunk swing.

In [123] fuzzy logic controllers were used for path planning, ditch crossing, and turning for a hexapod. The controllers processed ditch distance and angle information, and outputted actions for each leg. Initially, the rules were specified by the author, but it was assumed that a subset could be more effective. To achieve this, GAs were used to prune the rules, using a fitness function developed to minimize travel time and maximize walking efficiency. Through simulation it was found that, whatever the composition of the original rule set, the GA always improved the fitness.

The approach was extended in [122] to control a biped walking up stairs. Two fuzzy logic controllers were used, one to analyse the current system state, and one to prescribe actions based on this state. A ZMP based criterion was employed to monitor the success in creating an effective climbing gait. An author specified rule set and GA pruned subset were compared as in [123], plus a fully GA determined set. The fully GA specified set was found to be superior to the GA pruned set, which in turn was superior to the author set.

A biped ditch-crossing fuzzy logic algorithm was developed in [103] (also mentioned in Sections 2 and 4), where the gait was largely constructed analytically, but two fuzzy logic controllers were used for four free parameters. The rule sets were specified by GA and the best solution was found to be more stable and efficient than a fully analytical approach, but slightly worse than a NN based one.

Linguistic rules and data based rules were fused in [119] to produce bipedal control, where an expert based fuzzy rules system was combined with a data based neuro-fuzzy network to produce a final control

value. In [120, 121] an actor-critic system was used for on-line learning. The actor and critic were constructed using neuro-fuzzy networks, and had layers with neurons corresponding to system states. The architecture allowed learning algorithms to be applied to adjust the parameters of the fuzzy system. The classification of the fuzzy system was author specified and therefore the behaviour was interpretable. An external error reinforcement signal with fuzzy encoding was also used, again allowing expert knowledge to be incorporated and this way reducing the training time considerably.

In [122] comparisons were made for the GA fuzzy technique against others. The authors cited the ability to optimize the system as preferable over more restrictive analytical techniques. Compared to NNs, the authors valued the ease of interpretation of the rule based system and argued their implementation (using a GA) had better search of the solution space than the one in [120]. However, it should be noted that a GA version of [120] was presented in [121].

### 6.3 Rule Based Advantages and Disadvantages

Advantages:

- They can be very simple to implement;
- The rules are easily interpretable. Therefore, rule based systems can easily be seeded with expert knowledge;
- The flat nature of the rule tables makes them easy to optimize;
- Fuzzy systems can cope with varied environments.

Disadvantages:

- Transition tables are generally only good for specifying coarse movements;

- The methods are over constrained and therefore have less potential for optimisation than the other techniques.

## 7 Conclusion

The suggestions in this section, although following informed arguments, are hampered by a lack of research publications directly comparing different techniques. This section could be used as a starting point but obviously there are always other methods to try.

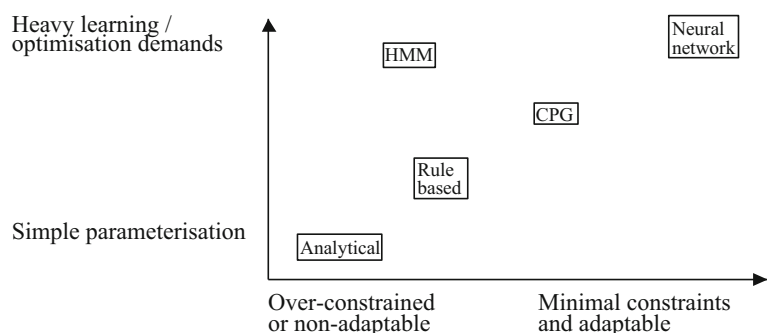
Despite being relatively scarce, some direct comparisons have been made between different approaches:

- A simple oscillator CPG was found to produce faster movement than a ZMP inverted pendulum model method in [46];
- A NN approach appeared to be more stable and efficient than a fuzzy logic approach for a ditch crossing robot in [103]. The fuzzy logic method was in turn found to be better than a ZMP IPM technique.

A continuum of approaches exists to produce the correct movement, from those that rely fully on prior knowledge and require no training, to those that are very generic pattern generators that require intensive training (Fig. 1). A summary of the highlighted advantages and disadvantages is given in Table 3.

Researchers' preference must also influence the decision. Analytical techniques can become very complex in more demanding situations. However, extensions such as the CWS [32] show that the analytical approach is capable of dealing with difficult environments. Conversely, RNNs may be more

**Fig. 1** Constraint/adaptability and learning demands of the control methods presented visually. The figure gives an approximate scatterplot of method positions, to aid comparison





**Table 3** Advantages and disadvantages of the reviewed control methods, examining characteristics including ease of implementation, generality, emergent behaviour, need for learning/optimisation, and special features

	Advantages	Disadvantages
Analytical	<ul style="list-style-type: none"> <li>• Accurate analysis can produce immediate results</li> <li>• Training not required so implementation in hardware can be easier</li> <li>• Long research history</li> <li>• Easy to interpret solutions</li> </ul>	<ul style="list-style-type: none"> <li>• Usually very task specific</li> <li>• Generally task limited with no emergent behaviours</li> <li>• Approximate modelling can produce inaccuracy or inefficiency</li> <li>• Simplification of analysis through over-constraint can lead to sub-optimal solutions</li> <li>• Constraints can produce ‘un-natural’ gaits</li> </ul>
CPG	<ul style="list-style-type: none"> <li>• Limit cycles are resilient to perturbation</li> <li>• Emergent behaviours</li> <li>• Clear parameter structure, good for optimisation</li> <li>• Less constrained than analytical, possible that better solutions exist</li> </ul>	<ul style="list-style-type: none"> <li>• Calibration, training or evolution often required</li> <li>• Less easy to interpret solutions compared to analytical</li> <li>• Still more constrained than some other approaches</li> <li>• Non-cyclic behaviour difficult to implement</li> </ul>
Neural networks	<ul style="list-style-type: none"> <li>• Least constrained approach</li> <li>• Capable of good limit cycle behaviour, resilient to perturbation, and able to interpolate behaviours</li> <li>• Emergent, new behaviours possible</li> <li>• Suitable for training or optimisation</li> <li>• Some interpretation possible with FFNNs</li> </ul>	<ul style="list-style-type: none"> <li>• Training costs can be high</li> <li>• Difficult to interpret method of solutions (black box solutions)</li> </ul>
HMM	<ul style="list-style-type: none"> <li>• Used for imitation tasks</li> <li>• Can be used as an identifier to initiate reflex behaviours</li> </ul>	<ul style="list-style-type: none"> <li>• No emergent behaviour</li> <li>• Lack of resilience to perturbation</li> <li>• Difficult to interpret method of solutions</li> </ul>
Rule based	<ul style="list-style-type: none"> <li>• Often used with a database of skills</li> <li>• Can be very simple to implement</li> <li>• Easy to interpret solutions</li> <li>• Easy to optimise</li> <li>• Fuzzy systems can cope with varied environments</li> </ul>	<ul style="list-style-type: none"> <li>• Often only coarse movements can be specified</li> <li>• Over constrained so less optimisation potential</li> </ul>

difficult to develop than the analytical solutions for simple situations, but their generality means a solution should still be available. If a researcher is knowledgeable in one domain but not the other it may be worthwhile searching for solutions using those methods even if another approach is theoretically more suitable.

The first criterion for selecting an approach is the availability of prior knowledge. If none is available,

then an analytical approach is usually ruled out because of the need for template foot trajectories (analysis is used to provide stability constraints). If prior knowledge is available, and the researcher wishes to use it, then wholly generic approaches such as RNNs, generic CPGs and Markov chains could be ruled out. If the problem is well understood, then a primarily analytical approach may be appropriate. If

only a limited amount of prior knowledge is available, then this can still be used in customising the structure of the control or setting some parameter values.

In this review, CPGs and NNs have been found to offer the option to evolve or learn solutions via reinforcement or supervision. This frees the designer from having to specify too much, as would be the case for an analytical approach. NNs offer a more generic option (compared to the oscillator primitives of CPGs), with RNNs being the most generic (allowing a combination of internally generated and input driven patterns). However, this generality comes at the expense of more difficult training. Oscillator based CPGs provide a more constrained option that can help reduce the training load. Transition tables should be noted for the ease of training, but they are limited in the kind of information they can generate.

To a certain extent all approaches could allow for the development of multiple skills or gaits through the use of a database of parameter values (as is common in HMM approaches [117]). However, analytical approaches will need a new analysis for each skill. CPGs and RNNs have shown examples of emergent gaits and smooth skill transitions as a natural consequence of their limit cycle behaviour [72, 73, 102, 112].

Hidden Markov Model systems are used for observation and classification as well as learning and generation. They are therefore an option for applications of learning through imitation [117], and initiating reflexes [116].

Rule based systems such as transition tables or fuzzy logic controllers have the advantage of clearly interpretable processes, even if they are developed through an optimisation algorithm. The transition table method is suitable for simple situations, with the fuzzy logic controller being more fault tolerant and appropriate for more complex situations [103].

Reinforcement learning allows effective control to be learned in situations where uncertainty exists, where a lack of input-output data prevents the use of supervised learning. However, reinforcement learning can often be slow, which often forces learning to be through computer simulation rather than in the real world. Systems that allow a hybrid of expert knowledge plus reinforcement learning give the option of speeding up the process by giving the system a head start. Options include RBFNs and fuzzy logic

**Table 4** Number of simulations used in evolving solutions using genetic algorithms, in a sample of publications covering rule based, CPG and neural network control methods

Reference	Method	Simulations
[103]	Rule based	900
[124]	Rule based	1150
[122]	Rule based	7200
[123]	Rule based	5000
[97]	CPG	500
[91]	CPG	5000
[105]	Neural network	40000

controllers, where networks or rule based systems are sufficiently interpretable in a sense that an expert can specify an initial solution, as investigated in [107, 123].

Although the less constrained methods potentially offer a greater range of solutions, their parameterisation can become complicated. As an example, Table 4 shows the large number of simulations often required for genetic algorithm based solutions (a popular choice). The table does not show the amount of time per simulation and this can vary largely between tasks. However, it can readily be seen that a great amount of simulation time is often required when control methods are general and need extensive parameterisation.

## References

1. Ha, T., Choi, C.-H.: An effective trajectory generation method for bipedal walking. *Robot. Auton. Syst.* **55**(10), 795–810 (2007)
2. Choi, Y., Kim, D., Oh, Y., You, B.-J.: Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion. *IEEE Trans. Robot.* **23**(6), 1285–1293 (2007)
3. Morimoto, J., Endo, G., Nakanishi, J., Cheng, G.: A biologically inspired biped locomotion strategy for humanoid robots: modulation of sinusoidal patterns by a coupled oscillator model. *IEEE Trans. Robot.* **24**(1), 185–191 (2008)
4. Reil, T., Husbands, P.: Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Trans. Evol. Comput.* **6**(2), 159–168 (2002)
5. Xiao, J., Su, J., Cheng, Y., Wang, F., Xu, X.: Research on gait planning of artificial leg based on central pattern generator. In: *Chinese Control and Decision Conference. CCDC 2008*, pp. 2147–2151 (2008)

6. Nandi, G.C., Ijspeert, A.J., Chakraborty, P., Nandi, A.: Development of Adaptive Modular Active Leg (AMAL) using bipedal robotics technology. *Robot. Auton. Syst.* **57**(6–7), 603–616 (2009)
7. Zordan, V.B., Majkowska, A., Chiu, B., Fast, M.: Dynamic response for motion capture animation. *ACM Trans. Graph.* **24**(3), 697–701 (2005)
8. Ijspeert, A.J., Hallam, J., Willshaw, D.: Evolving swimming controllers for a simulated lamprey with inspiration from neurobiology. *Adapt. Behav.* **7**(2), 151 (1999)
9. Hirukawa, H., Kanehiro, F., Kaneko, K., Kajita, S., Fujiwara, K., Kawai, Y., Tomita, F., Hirai, S., Tanie, K., Isozumi, T., Akachi, K., Kawasaki, T., Ota, S., Yokoyama, K., Handa, H., Fukase, Y., ichiro Maeda, J., Nakamura, Y., Tachi, S., Inoue, H.: Humanoid robotics platforms developed in HRP. *Robot. Auton. Syst.* **48**(4), 165–175 (2004)
10. Lachat, D., Crespi, A., Ijspeert, A.J.: BoxyBot: a swimming and crawling fish robot controlled by a central pattern generator. In: *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics, BioRob 2006*, pp. 643–648 (2006)
11. Watanabe, K., Tajima, A., Izumi, K.: Locomotion pattern generation of semi-looper type robots using central pattern generators based on van der Pol oscillators. In: *6th IEEE International Conference on Industrial Informatics, INDIN 2008*, pp. 377–382 (2008)
12. Goswami, D., Vadakkepat, P.: Planar bipedal jumping gaits with stable landing. *IEEE Trans. Robot.* **25**(5), 1030–1046 (2009)
13. Hosoda, K., Takuma, T., Nakamoto, A., Hayashi, S.: Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion. *Robot. Auton. Syst.* **56**(1), 46–53 (2008)
14. Braun, D.J., Goldfarb, M.: A control approach for actuated dynamic walking in biped robots. *IEEE Trans. Robot.* **25**(6), 1292–1303 (2009)
15. Asano, F., Yamakita, M., Kamamichi, N., Luo, Z.-W.: A novel gait generation for biped walking robots based on mechanical energy constraint. *IEEE Trans. Robot. Autom.* **20**(3), 565–573 (2004)
16. Hirukawa, H., Hattori, S., Kajita, S., Harada, K., Kaneko, K., Kanehiro, F., Morisawa, M., Nakaoka, S.: A Pattern Generator of Humanoid Robots Walking on a Rough Terrain. In: *2007 IEEE International Conference on Robotics and Automation*, pp. 2181–2187 (2007)
17. Miossec, S., Aoustin, Y.: A simplified stability study for a biped walk with underactuated and overactuated phases. *Int. J. Robot. Res.* **24**(7), 551 (2005)
18. Chevallereau, C., Westervelt, E., Grizzle, J.: Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. *Int. J. Robot. Res.* **24**(6), 464 (2005)
19. Asano, F., Luo, Z.-W., Yamakita, M.: Biped Gait Generation and Control Based on a Unified Property of Passive Dynamic Walking. *IEEE Trans. Robot.* **21**(4), 754–762 (2005)
20. Reisinger, K.D., Moskowicz, G.D.: Bipedal locomotion: stopping and the standing/balance gait. *Int. J. Robot. Res.* **18**(3), 333 (1999)
21. Tlalolini, D., Chevallereau, C., Aoustin, Y.: Comparison of different gaits with rotation of the feet for a planar biped. *Robot. Auton. Syst.* **57**(4), 371–383 (2009)
22. Harada, K., Kajita, S., Kaneko, K., Hirukawa, H.: Dynamics and balance of a humanoid robot during manipulation tasks. *IEEE Trans. Robot.* **22**(3), 568–575 (2006)
23. Asano, F., Luo, Z.-W.: Energy-efficient and high-speed dynamic biped locomotion based on principle of parametric excitation. *IEEE Trans. Robot.* **24**(6), 1289–1301 (2008)
24. Kim, Y.-D., Lee, B.-J., Ryu, J.-H., Kim, J.-H.: Landing force control for humanoid robot by time-domain passivity approach. *IEEE Trans. Robot.* **23**(6), 1294–1301 (2007)
25. Lee, B.-J., Stonier, D., Kim, Y.-D., Yoo, J.-K., Kim, J.-H.: Modifiable walking pattern of a humanoid robot by using allowable zmp variation. *IEEE Trans. Robot.* **24**(4), 917–925 (2008)
26. Czarnetzki, S., Kerner, S., Urbann, O.: Observer-based dynamic walking control for biped robots. *Robot. Auton. Syst.* **57**(8), 839–845 (2009)
27. Wieber, P.-B., Chevallereau, C.: Online adaptation of reference trajectories for the control of walking systems. *Robot. Auton. Syst.* **54**(7), 559–566 (2006)
28. Sugihara, T., Nakamura, Y., Inoue, H.: Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control. *Int. J. Robot. Res.* **21**(12), 1404–1409 (2002)
29. Seipel, J.E., Holmes, P.: Running in three dimensions: analysis of a point-mass sprung-leg model. *Int. J. Robot. Res.* **24**(8), 674 (2005)
30. Vukobratovic, M., Borovac, B.: Zero-moment point—thirty five years of its life. *Int. J. Humanoid Robot.* **1**(1), 157–173 (2004)
31. Park, J.H.: Fuzzy-logic zero-moment-point trajectory generation for reduced trunk motions of biped robots. *Fuzzy Sets Syst.* **134**(1), 189–203 (2003)
32. Hirukawa, H., Hattori, S., Harada, K., Kajita, S., Kaneko, K., Kanehiro, F., Fujiwara, K., Morisawa, M.: A universal stability criterion of the foot contact of legged robots - adios ZMP. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006*, pp. 1976–1983 (2006)
33. Koyanagi, K., Hirukawa, H., Hattori, S., Morisawa, M., Nakaoka, S., Harada, K., Kajita, S.: A pattern generator of humanoid robots walking on a rough terrain using a handrail. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, pp. 2617–2622 (2008)
34. Hyon, S.-H.: Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. *IEEE Trans. Robot.* **25**(1), 171–178 (2009)
35. Mitobe, K., Capi, G., Nasu, Y.: Control of walking robots based on manipulation of the zero moment point. *Robotica* **18**(06), 651–657 (2001)
36. Furuta, T., Tawara, T., Okumura, Y., Shimizu, M., Tomiyama, K.: Design and construction of a series of compact humanoid robots and development of biped walk control strategies. *Robot. Auton. Syst.* **37**(2–3), 81–100 (2001)
37. Kajita, S., Yamaura, T., Kobayashi, A.: Dynamic walking control of a biped robot along a potential energy conserving orbit. *IEEE Trans. Robot. Autom.* **8**(4), 431–438 (1992)
38. Zheng, Y.F., Shen, J.: Gait synthesis for the SD-2 biped robot to climb sloping surface. *IEEE Trans. Robot. Autom.* **6**(1), 86–96 (1990)

39. Dong, H., Zhao, M.G., Zhang, J., Zhang, N.Y.: Hardware design and gait generation of humanoid soccer robot Stepper-3D. *Robot. Auton. Syst.* **57**(8), 828–838 (2009)
40. Kagami, S., Mochimaru, M., Ehara, Y., Miyata, N., Nishiwaki, K., Kanade, T., Inoue, H.: Measurement and comparison of humanoid H7 walking with human being. *Robot. Auton. Syst.* **48**(4), 177–187 (2004)
41. Kanehiro, F., Hirukawa, H., Kajita, S.: Openhrp: Open architecture humanoid robotics platform. *Int. J. Robot. Res.* **23**(2), 155 (2004)
42. Ude, A., Atkeson, C.G., Riley, M.: Programming full-body movements for humanoid robots by observation. *Robot. Auton. Syst.* **47**(2–3), 93–108 (2004)
43. Yoo, J.-K., Lee, B.-J., Kim, J.-H.: Recent progress and development of the humanoid robot HanSaRam. *Robot. Auton. Syst.* **57**(10), 973–981 (2009)
44. Ono, K., Takahashi, R., Shimada, T.: Self-excited walking of a biped mechanism. *Int. J. Robot. Res.* **20**(12), 953 (2001)
45. Ono, K., Furuichi, T., Takahashi, R.: Self-excited walking of a biped mechanism with feet. *Int. J. Robot. Res.* **23**(1), 55 (2004)
46. Vadakkepat, P., Sin, N.B., Goswami, D., Zhang, R.X., Tan, L.Y.: Soccer playing humanoid robots: Processing architecture, gait generation and vision system. *Robot. Auton. Syst.* **57**(8), 776–785 (2009)
47. Herr, H.M., McMahon, T.A.: A trotting horse model. *Int. J. Robot. Res.* **19**(6), 566 (2000)
48. Herr, H.M., McMahon, T.A.: A galloping horse model. *Int. J. Robot. Res.* **20**(1), 26 (2001)
49. Formal'sky, A., Chevellereau, C., Perrin, B.: On ballistic walking locomotion of a quadruped. *Int. J. Robot. Res.* **19**(8), 743–761 (2000)
50. Poulakakis, I., Smith, J.A., Buehler, M.: Modeling and Experiments of Untethered Quadrupedal Running with a Bounding Gait: The Scout II Robot. *Int. J. Robot. Res.* **24**(4), 256 (2005)
51. Garcia, E., Gonzalez de Santos, P.: On the improvement of walking performance in natural environments by a compliant adaptive gait. *IEEE Trans. Robot.* **22**(6), 1240–1253 (2006)
52. Albiez, J.C., Luksch, T., Berns, K., Dillmann, R.: Reactive reflex-based control for a four-legged walking machine. *Robot. Auton. Syst.* **44**(3–4), 181–189 (2003)
53. Raibert, M., Chepponis, M., Brown, H. Jr.: Running on four legs as though they were one. *IEEE J. Robot. Autom.* **2**(2), 70–82 (1986)
54. Lee, T.-T., Liao, C.-M., Chen, T.K.: On the stability properties of hexapod tripod gait. *IEEE J. Robot. Autom.* **4**(4), 427–434 (1988)
55. Altendorfer, R., Koditschek, D.E., Holmes, P.: Stability analysis of legged locomotion models by symmetry-factored return maps. *Int. J. Robot. Res.* **23**(10–11), 979 (2004)
56. Bessonnet, G., Seguin, P., Sardain, P.: A parametric optimization approach to walking pattern synthesis. *Int. J. Robot. Res.* **24**(7), 536 (2005)
57. Chevallereau, C., Grizzle, J.W., Shih, C.-L.: Asymptotically stable walking of a five-link underactuated 3-d bipedal robot. *IEEE Trans. Robot.* **25**(1), 37–50 (2009)
58. Westervelt, E.R., Grizzle, J.W., Koditschek, D.E.: Hybrid zero dynamics of planar biped walkers. *IEEE Trans. Autom. Control* **48**(1), 42–56 (2003)
59. Westervelt, E., Buche, G., Grizzle, J.: Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *Int. J. Robot. Res.* **23**(6), 559 (2004)
60. Shapiro, A., Rimon, E., Ohev-Zion, A.: On the mechanics of natural compliance in frictional contacts and its effect on grasp stiffness and stability. *Int. J. Robot. Res.* **32**(4), 425–445 (2013)
61. Hemker, T., Stelzer, M., von Stryk, O., Sakamoto, H.: Efficient Walking Speed Optimization of a Humanoid Robot. *Int. J. Robot. Res.* **28**(2), 303–314 (2009)
62. Geva, Y., Shapiro, A.: A combined potential function and graph search approach for free gait generation of quadruped robots. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 5371–5376. IEEE (2012)
63. Pal, P.K., Jayarajan, K.: Generation of free gait—a graph search approach. *IEEE Trans. Robot. Autom.* **7**(3), 299–305 (1991)
64. Sznajder, M., Damborg, M.J.: An adaptive controller for a one-legged mobile robot. *IEEE Trans. Robot. Autom.* **5**(2), 253–259 (1989)
65. Capi, G., Nasu, Y., Barolli, L.: Application of Genetic Algorithms for biped robot gait synthesis optimization during walking and going up-stairs. *Adv. Robot.* **15**(6), 675–694 (2001)
66. Capi, G., Kaneko, S., Mitobe, K., Barolli, L., Nasu, Y.: Optimal trajectory generation for a prismatic joint biped robot using genetic algorithms. *Robot. Auton. Syst.* **38**(2), 119–128 (2002)
67. Capi, G., Nasu, Y., Barolli, L., Mitobe, K.: Real time gait generation for autonomous humanoid robots: A case study for walking. *Robot. Auton. Syst.* **42**(2), 107–116 (2003)
68. Feng, K., Chew, C.-M., Hong, G.-S., Zielinska, T.: Bipedal locomotion control using a four-compartmental central pattern generator. In: 2005 IEEE International Conference on Mechatronics and Automation, vol. 1513, pp. 1515–1520, (2005)
69. Komatsu, T., Usui, M.: Dynamic walking and running of a bipedal robot using hybrid central pattern generator method. In: 2005 IEEE International Conference on Mechatronics and Automation, vol. 982, pp. 987–992 (2005)
70. Yuasa, H., Ito, M.: A Theory on Autonomous Distributed Systems with Application to a Gait Pattern Generator of Quadruped. In: American Control Conference, pp. 2268–2273 (1991)
71. Nakada, K., Asai, T., Amemiya, Y.: An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion. *IEEE Trans. Neural Netw.* **14**(5), 1356–1365 (2003)
72. Asa, K., Ishimura, K., Wada, M.: Behavior transition between biped and quadruped walking by using bifurcation. *Robot. Auton. Syst.* **57**(2), 155–160 (2009)
73. Bay, J.S., Hemami, H.: Modeling of a neural pattern generator with coupled nonlinear oscillators. *IEEE Trans. Biomed. Eng. BME* **34**(4), 297–306 (1987)

74. Righetti, L., Ijspeert, A.J.: Pattern generators with sensory feedback for the control of quadruped locomotion. In: IEEE International Conference on Robotics and Automation. ICRA 2008, pp. 819–824 (2008)
75. Fukuoka, Y., Kimura, H., Cohen, A.H.: Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *Int. J. Robot. Res.* **22**(3–4), 187 (2003)
76. Arena, P., Fortuna, L., Frasca, M., Patane, L.: CNN based central pattern generators with sensory feedback. In: Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications. (CNNA 2002), pp. 275–282 (2002)
77. Feng, H., Wang, R.: Construction of central pattern generator for quadruped locomotion control. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. AIM 2008, pp. 979–984 (2008)
78. Liu, C., Chen, Q., Zhang, J.: Coupled Van Der Pol oscillators utilised as Central pattern generators for quadruped locomotion. In: Control and Decision Conference. CCDC '09, Chinese, pp. 3677–3682 (2009)
79. Takemura, H., Deguchi, M., Ueda, J., Matsumoto, Y., Ogasawara, T.: Slip-adaptive walk of quadruped robot. *Robot. Auton. Syst.* **53**(2), 124–141 (2005)
80. Micci-Barreca, D., Ogmen, H.: A central pattern generator for insect gait production. In: From Perception to Action Conference. Proceedings, pp. 348–351 (1994)
81. Klaassen, B., Linnemann, R., Spenneberg, D., Kirchner, F.: Biomimetic walking robot SCORPION: Control and modeling. *Robot. Auton. Syst.* **41**(2–3), 69–76 (2002)
82. Zhao, W., Yu, J., Fang, Y., Wang, L.: Development of Multi-mode Biomimetic Robotic Fish Based on Central Pattern Generator. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3891–3896 (2006)
83. McIsaac, K.A., Ostrowski, J.P.: Experimental verification of open-loop control for an underwater eel-like robot. *Int. J. Robot. Res.* **21**(10–11), 849 (2002)
84. Mehta, V., Brennan, S., Gandhi, F.: Experimentally verified optimal serpentine gait and hyperredundancy of a rigid-link snake robot. *IEEE Trans. Robot.* **24**(2), 348–360 (2008)
85. Ijspeert, A.J., Crespi, A.: Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In: IEEE International Conference on Robotics and Automation, pp. 262–268 (2007)
86. Righetti, L., Ijspeert, A.J.: Programmable central pattern generators: an application to biped locomotion control. In: Proceedings 2006 IEEE International Conference on Robotics and Automation. ICRA 2006, pp. 1585–1590 (2006)
87. Geng, T., Porr, B., Worotter, F.: Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning. *Int. J. Robot. Res.* **25**(3), 259 (2006)
88. Endo, G., Morimoto, J., Matsubara, T., Nakanishi, J., Cheng, G.: Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. *Int. J. Robot. Res.* **27**(2), 213–228 (2008)
89. Hliot, R., Espiau, B.: Online generation of cyclic leg trajectories synchronized with sensor measurement. *Robot. Auton. Syst.* **56**(5), 410–421 (2008)
90. Rutishauser, S., Sprowitz, A., Righetti, L., Ijspeert, A.J.: Passive compliant quadruped robot using Central Pattern Generators for locomotion control. In: 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechanics. BioRob 2008, pp. 710–715 (2008)
91. Tanev, I., Ray, T., Buller, A.: Automated Evolutionary Design, Robustness, and Adaptation of Sidewinding Locomotion of a Simulated Snake-Like Robot. *IEEE Trans. Robot.* **21**(4), 632–645 (2005)
92. Shan, J., Junshi, C., Jiapin, C.: Design of central pattern generator for humanoid robot walking based on multi-objective GA. In: 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2000), Proceedings, vol. 1933, pp. 1930–1935 (2000)
93. Kim, J.-J., Lee, J.-J.: Gait adaptation method of biped robot for various terrains using central pattern generator (CPG) and learning mechanism. In: International Conference on Control, Automation and Systems. ICCAS '07, pp. 10–14 (2007)
94. Wolff, K., Pettersson, J., Heralic, A., Wahde, M.: Structural Evolution of Central Pattern Generators for Bipedal Walking in 3D Simulation. In: IEEE International Conference on Systems, Man and Cybernetics. SMC '06, pp. 227–234 (2006)
95. Inada, H., Ishii, K.: Behavior generation of bipedal robot using central pattern generator(CPG) (1st report: CPG parameters searching method by genetic algorithm). In: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2003), Proceedings, vol. 2173, pp. 2179–2184 (2003)
96. Russell, A., Orchard, G., Etienne-Cummings, R.: Configuring of Spiking Central Pattern Generator Networks for Bipedal Walking Using Genetic Algorithms. In: IEEE International Symposium on Circuits and Systems. ISCAS 2007, pp. 1525–1528 (2007)
97. Hornby, G.S., Takamura, S., Yamamoto, T., Fujita, M.: Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Trans. Robot.* **21**(3), 402–410 (2005)
98. Akio, S., Masaki, Y.: Design of a novel central pattern generator and the hebbian motion learning. In: Control Applications, (CCA) & Intelligent Control. (ISIC) 2009, IEEE. pp. 1655–1660 (2009)
99. Durr, V., Krause, A.F., Schmitz, J., Cruse, H.: Neuroethological concepts and their transfer to walking machines. *Int. J. Robot. Res.* **22**(3–4), 151 (2003)
100. Manoonpong, P., Wörgötter, F.: Efference copies in neural control of dynamic biped walking. *Robot. Auton. Syst.* **57**(11), 1140–1153 (2009)
101. Xia, Y., Wang, J., Fok, L.-M.: Grasping-force optimization for multifingered robotic hands using a recurrent neural network. *IEEE Trans. Robot. Autom.* **20**(3), 549–554 (2004)
102. Srinivasan, S., Gander, R.E., Wood, H.C.: A movement pattern generator model using artificial neural networks. *IEEE Trans. Biomed. Eng.* **39**(7), 716–722 (1992)
103. Vundavilli, P.R., Pratihari, D.K.: Dynamically balanced optimal gaits of a ditch-crossing biped robot. *Robot. Auton. Syst.* **58**(4), 349–361 (2010)
104. Gallagher, J.C., Beer, R.D., Espenschied, K.S., Quinn, R.D.: Application of evolved locomotion controllers to a hexapod robot. *Robot. Auton. Syst.* **19**(1), 95–103 (1996)

105. Beer, R.D., Gallagher, J.C.: Evolving dynamical neural networks for adaptive behavior. *Adapt. Behav.* **1**(1), 91–122 (1992)
106. Petridis, V., Papaikonomou, A.: Recurrent neural networks as pattern generators. In: 1994 IEEE International Conference on Neural Networks. IEEE World Congress on Computational Intelligence, vol.872, pp. 872–875 (1994)
107. Ilg, W., Berns, K.: A learning architecture based on reinforcement learning for adaptive control of the walking machine LAURON. *Robot. Auton. Syst.* **15**(4), 321–334 (1995)
108. Ilg, W., Berns, K., Mhlfriedel, T., Dillmann, R.: Hybrid learning concepts based on self-organizing neural networks for adaptive control of walking machines. *Robot. Auton. Syst.* **22**(3–4), 317–327 (1997)
109. Benbrahim, H., Franklin, J.A.: Biped dynamic walking using reinforcement learning. *Robot. Auton. Syst.* **22**(3–4), 283–302 (1997)
110. Sabourin, C., Bruneau, O., Buche, G.: Control strategy for the robust dynamic walk of a biped robot. *Int. J. Robot. Res.* **25**(9), 843–860 (2006)
111. Sabourin, C., Bruneau, O.: Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. *Robot. Auton. Syst.* **51**(2–3), 81–99 (2005)
112. Wyffels, F., Schrauwen, B.: Design of a Central Pattern Generator Using Reservoir Computing for Learning Human Motion. In: *Advanced Technologies for Enhanced Quality of Life. AT-EQUAL '09*, pp. 118–122 (2009)
113. Berns, K., Dillmann, R., Piekenbrock, S.: Neural networks for the control of a six-legged walking machine. *Robot. Auton. Syst.* **14**(2–3), 233–244 (1995)
114. Inamura, T., Toshima, I., Tanie, H., Nakamura, Y.: Embodied symbol emergence based on mimesis theory. *Int. J. Robot. Res.* **23**(4–5), 363 (2004)
115. Lee, D., Nakamura, Y.: Mimesis model from partial observations for a humanoid robot. *Int. J. Robot. Res.* **29**(1), 60 (2010)
116. Hohn, O., Gerth, W.: Probabilistic balance monitoring for bipedal robots. *Int. J. Robot. Res.* **28**(2), 245–256 (2009)
117. Kulic, D., Nakamura, Y.: Incremental learning and memory consolidation of whole body human motion primitives. *Adapt. Behav.* **17**(6), 484 (2009)
118. Kulic, D., Takano, W., Nakamura, Y.: Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *Int. J. Robot. Res.* **27**(7), 761–784 (2008)
119. Zhou, C., Ruan, D.: Integration of linguistic and numerical information for biped control. *Robot. Auton. Syst.* **28**(1), 53–70 (1999)
120. Zhou, C., Meng, Q.: Dynamic balance of a biped robot using fuzzy reinforcement learning agents. *Fuzzy Sets Syst.* **134**(1), 169–187 (2003)
121. Zhou, C.: Robot learning with GA-based fuzzy reinforcement learning agents. *Inf. Sci.* **145**(1–2), 45–68 (2002)
122. Jha, R.K., Singh, B., Pratihari, D.K.: On-line stable gait generation of a two-legged robot using a genetic-fuzzy system. *Robot. Auton. Syst.* **53**(1), 15–35 (2005)
123. Pratihari, D.K., Deb, K., Ghosh, A.: Optimal path and gait generations simultaneously of a six-legged robot using a GA-fuzzy approach. *Robot. Auton. Syst.* **41**(1), 1–20 (2002)
124. Barfoot, T.D., Earon, E.J.P., D'Eleuterio, G.M.T.: Experiments in learning distributed control for a hexapod robot. *Robot. Auton. Syst.* **54**(10), 864–872 (2006)
125. Pal, P.K., Kar, D.C.: Gait optimization through search. *Int. J. Robot. Res.* **19**(4), 394 (2000)
126. McIsaac, K.A., Ostrowski, J.P.: Motion planning for anguilliform locomotion. *IEEE Trans. Robot. Autom.* **19**(4), 637–652 (2003)
127. Goswami, A.: Postural stability of biped robots and the foot-rotation indicator (FRI) point. *Int. J. Robot. Res.* **18**(6), 523 (1999)
128. Popovic, M.B., Goswami, A., Herr, H.: Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *Int. J. Robot. Res.* **24**(12), 1013 (2005)
129. Grillner, S.: Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiol. Rev.* **55**(2), 247 (1975)
130. Marder, E., Bucher, D.: Central pattern generators and the control of rhythmic movements. *Curr. Biol.* **11**(23), R986–R996 (2001)
131. Duysens, J., Van de Crommert, H.W.A.A.: Neural control of locomotion; Part 1: The central pattern generator from cats to humans. *Gait & Posture* **7**(2), 131–141 (1998)
132. MacKay-Lyons, M.: Central pattern generation of locomotion: a review of the evidence. *Phys. Ther.* **82**(1), 69 (2002)
133. Syed, N., Bulloch, A., Lukowiak, K.: In vitro reconstruction of the respiratory central pattern generator of the mollusk *Lymnaea*. *Science* **250**(4978), 282 (1990)
134. Zhu, K., Zhang, D., Lan, L.: On Central Pattern Generator of Biological Motor System. In: 9th International Conference on Control, Automation, Robotics and Vision. ICARCV '06, pp. 1–5 (2006)
135. Wu, Q., Liu, C., Zhang, J., Chen, Q.: Survey of locomotion control of legged robots inspired by biological concept. *Sci. China Ser. F: Inf. Sci.* **52**(10), 1715–1729 (2009)
136. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: A review. *Neural Netw.* **21**(4), 642–653 (2008)
137. Grillner, S., Deliagina, T., Manira, A.E., Hill, R.H., Orlovsky, G.N., Walln, P., Ekeberg, O., Lansner, A.: Neural networks that co-ordinate locomotion and body orientation in lamprey. *Trends Neurosci.* **18**(6), 270–279 (1995)
138. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol. Cybern.* **52**(6), 367–376 (1985)
139. Lee, W.M., Yam, Y.: Construction of Central Pattern Generator Using Piecewise Affine Systems. In: IEEE International Conference on Control and Automation. ICCA 2007, pp. 1729–1734 (2007)
140. Grossberg, S., Pribe, C., Cohen, M.A.: Neural control of interlimb oscillations. *Biol. Cybernet.* **77**, 131–140 (1997)
141. Zhang, D., Hu, D., Shen, L., Xie, H.: Design of a Central Pattern Generator for Bionic-robot Joint with Angular Frequency Modulation. In: IEEE International Conference on Robotics and Biomimetics. ROBIO '06, pp. 1664–1669 (2006)

142. Albus, J.S.: A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *J. Dyn. Syst. Meas. Control.* **97**(3), 220–227 (1975)
143. Rumelhart, D.E., Hintont, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986)
144. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Stat.* **41**(1), 164–171 (1970)
145. Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **13**(2), 260–269 (1967)