Gait Adaptation in a Quadruped Robot

M. Anthony Lewis
Iguana Robotics, Inc.
tlewis@iguana-robotics.com

Please Send All Correspondences to:

M. Anthony Lewis
Iguana Robotics, Inc.
P.O. Box 628
Mahomet, IL 61853, USA

e-mail: tlewis@iguana-robotics.com

Running Header:

Gait Adaptation

Keywords:

Walking Machines
Central Pattern Generators
Neural Networks
Legged Locomotion
Non-linear oscillators
Biologically inspired robots

Abstract

A newborn foal can learn to walk soon after birth through a process of rapid adaptation acting on its locomotor controller. It is proposed here that this kind of adaptation can be modeled as a distributed system of adaptive modules (AMs) acting on a distributed system of adaptive oscillators called Adaptive Ring Rules (ARRs), augmented with appropriate and simple reflexes. It is shown that such a system can self-program through interaction with the environment. The adaptation emerges spontaneously as several discrete stages: Body twisting, short quick steps, and finally longer, coordinated stepping.

This approach is demonstrated on a quadrupedal robot. The result is that the system can learn to walk several minutes after inception.

1.0 Introduction

A few minutes after birth, a foal can walk and then run. It is remarkable that the animal learns to coordinate the many muscles of the legs and trunk in such a short period of time. It is not likely that any learning algorithm could program a nervous system *ab initio* with so few training epochs. Nor is it likely that the foal's locomotor controller is completely determined before birth. How can this ability be explained? How can this ability be incorporated into the control system of a walking machine?

Researchers in biology have presented clear evidence of a functional unit of the central nervous system, the Central Pattern Generator (CPG), which can cause rhythmic movement of the trunk and limb muscles (Grillner and Wallén 1985). In adult animals, the output of these cells can generate muscle activity that is very similar to activity during normal walking, even when sensory feedback has been eliminated (Grillner and Zangger 1975). The CPG begins its activity before birth, although its activity does not appear to imitate the details of a particular walking animal, it is apparently correlated with the animal's class, i.e. amphibian, reptile, mammal, etc. (Bekoff 1985; Cohen 1988). Apparently, the basic structure of the CPG network is laid down by evolution. How is this basic structure adapted to produce the detailed coordination needed to control a walking animal?

The answer to this question is important to robotics for the following reason. CPGs have been well studied as a basic coordinating mechanism (Cohen et al. 1982; Bay and Hemami 1987; Matsuoka 1987; Rand et al. 1988; Taga et al. 1991; Collins and Stewart 1993; Murray 1993; Zielinska 1996; Jalics et al. 1997; Ito et al.; Kimura et al. 1999). However, the details of how this system can automatically adapt to control a real robot are not clear. A good goal would be to describe a general strategy for matching a generic CPG to a particular robot in real-time, with a minimal amount of interaction with the environment.

Reinforcement learning has been applied on long time scales to certain problems in walking (learning coordination and basic leg movement) (Ilg and Berns 1995), but the time scales of such approach is too long to explain the quick learning of animals just after birth.

The author suggests that part of the answer may be in the use of a number of simple innate internal models to evaluate the performance of the rapidly developing nervous system. These *innate internal models* could be used to adaptively tune CPGs during phases of rapid development. Fig. 3 illustrates the training concept. A CPG generates a signal destined for a group of actuators (muscle) as well as a second signal, which is a *copy* of the signal sent to the actuators destined for an innate forward model. In biology, a copy of the motor signal is called an efference copy (Sperry 1950). A forward model as described by Kawato (Kawato and Wolpert 1998) is a functional model of the forward dynamics of the system. We use very simplified, innate forward models. These forward models predict the sensory expectation, or the *desired* consequence of CPG activity. This information is compared, and an adaptive rule then modifies the CPG.

The author suggests that a handful of adaptive mechanisms may be used for rapid tuning of a generic CPG. For example, the adaptive model can be used to ensure coordination of limbs with the environment.

The use of simplified, innate models is the most conservative stance possible. The intent is to make the fewest assumptions possible about the 'knowledge' that the nervous system has about the body that it is trying to control. By demonstrating how this process may be used in a physical device—a robot—we give compelling evidence that this approach is sufficient.

This article reports on an investigation into how a group of forward models could be used to adaptively tune a CPG in a real robot. As these models are innate, we assume they are simple; it would not be satisfying if these models were as complex as the behaviors that they help generate. Secondly, these models should not be detailed, accurate models of the forward dynamics of the robot. If innate models are used, their simplicity prohibits them from detailing accurate information about the structure of the pattern generator. The study here supposes that once the CPGs have been tuned to produce basic locomotion, other, more general learning mechanisms would take over to create a more refined gait. These learning mechanisms might include reinforcement learning or supervised learning methods.

2.0 Model

It is assumed that the CPG for the control of a robot could be divided into a network of functional units called unit CPGs (uCPGs). Referring to Figure 1, two uCPGs, denoted as Tw+ and Tw- are coupled together to produce the main coordinating unit in the robot. These uCPGs control trunk twist and will be important as the robot learns to walk. These uCPGs in turn coordinate with hip flexors. The hip flexors coordinate with hip extensors. The hip uCPGs coordinate with the knee uCPGs.

Output from the flexor and extensors are transformed into movement commands by an output function. The movement commands are then recombined to create knee and hip commands.

Certain key parameters can be adjusted to adapt the network to a particular robot. Section 2.1 gives a model of the uCPGs. Section 2.2 describes how motor commands are combined to create joint position commands. Section 2.3 describes built-in reflexes. Section 2.4 describes adaptive modules that adjust the key parameters of the network.

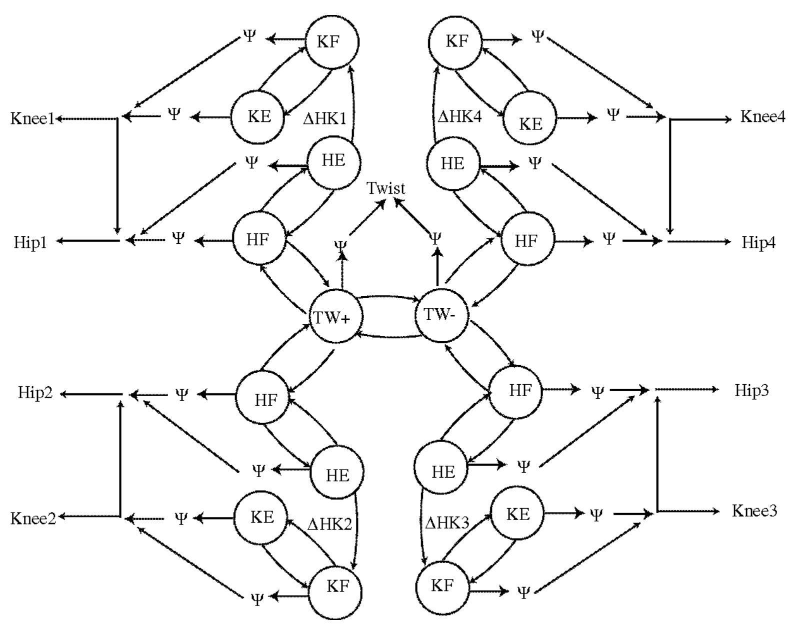


Figure 1. Organization of control system. The control system of the robot is represented as a network of uCPGs. Each circle above represents a uCPG. Connections, transmitting phase information, are shown as arrows between uCPGs. Each function Ψ converts phase information to a motor command. Motor commands are combined together to produce joint level commands for Hips and Knees.
Abbreviations. KF: Knee Flexor, KE: Knee Extensor, HE: Hip Extensor, HF: Hip Flexor, TW+: Positive Twist, TW-: Negative Twist

2.1 Modeling uCPGs using Adaptive Ring Rules

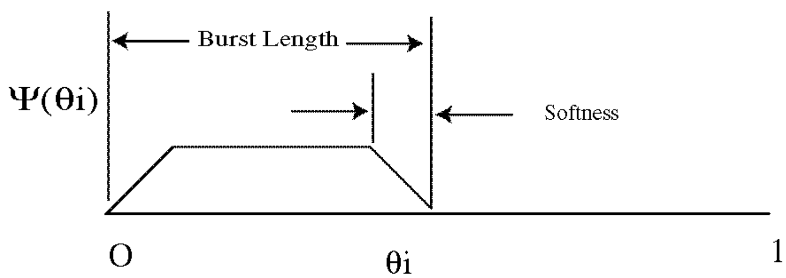


Figure 2. Parameters of the output function Ψ . Here we model the bursting of the uCPG as an average firing rate. The rate at which the firing rate increases/decreases is the softness; the greater the softness, the slower the rate. The burst duration is the percent of the uCPG period.

The uCPGs are modeled as a distributed system of coupled clocks (Murray 1993) with additional machinery to control a robot. This model of an uCPG is called an Adaptive Ring Rule (ARR). In earlier work we presented a general form of ARR (Lewis 1996). In this work, we use a specific coupling term inspired by (Ito et al. 1998):

$$\dot{\theta}_i = \omega_i + \sum\nolimits_{\forall j \neq i} (\theta_j - \theta_i - \Delta_{ij}) \cdot \alpha_{ij} \qquad (1)$$

$$y_i = \Psi(\theta_i, BL, Softness) \qquad (2)$$

$$x_i = y_i \cdot (1 - I_k) \cdot A_i$$
$$I_k \in [0,1] \qquad (3)$$

where $\theta_i \in S^1$ is the phase of the *ith* ARR in the system, ω_i is the nominal oscillator frequency, Δ_{ij} is the desired phase difference between the ith and jth ARRs, α_{ij} is the oscillator-oscillator coupling strength, the function $\Psi()$ is a mapping from the phase to the *ith* ARR output variable y_i. The function used in this work is a trapezoid. See Fig 2. The shape of the trapezoid is parameterized by BL, the 'burst length', or interval when the output is active, and $Softness$, the transition time (expressed in phase units) from high to low and low to high values. The output variable y_i is mapped to a motor command x_i by application of an inhibition term I_k, and a gain term A_i.

2.2 Motor Primitives

Many useful movements require a coordinated movement of several joints. For example, the flexing or extending of a leg can be considered a movement primitive. This movement requires the coordination of both knee and hip. The use of such motor primitives significantly simplifies control. It is assumed that a leg flexion and extension primitive is innate.

Given equal upper and lower limb lengths, the leg synergy is expressed as

$$\begin{bmatrix} x_{Hip_i} \\ x_{Knee_i} \end{bmatrix} = \begin{bmatrix} -.5 & .5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{flex_i} \\ x_{rotate_i} \end{bmatrix} + \begin{bmatrix} b_i \\ 0 \end{bmatrix} \qquad (4)$$

where x_{flex_i} and x_{rotate_i} are the output of ARRs generating leg flexion and hip rotation commands, respectively. x_{Hip_i} and x_{Knee_i} are joint position commands to actuators controlling the hip and knee joints.

Likewise, the robot can adjust its balance by leaning forward or backward. This is denoted as:

$$\begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \cdot L \qquad (5)$$

where L controls a coordinated 'lunge' movement backward or forward.

Sway (side to side rocking of the body) is controlled similarly (equation not shown). Here a variable Sw controls the side to side movement.

2.3 Reflexes

Reflexes are different than CPG-driven motor activity. In CPGs, activity is generated by an intrinsic oscillator. In a reflex, sensory input releases a behavior. Reflexes are included as part of a stabilizing mechanism. We consider two reflexes here: a paw extension reflex and a postural control reflex, which compensates for uneven distribution of weight.

2.3.1 Paw Extension Reflex

The paw extension reflex is crucial for enforcing leg support. If the leg is bearing weight, flexion in that leg is inhibited. This reflex is of particular importance at two moments in the leg's step cycle. If the leg ARR commands a leg flexion, this flexion will be temporarily inhibited until the leg is un-weighted. Without this reflex, the robot could topple over very easily. The second moment when this reflex is important is at the moment of contact of the leg with the ground. If the leg is still flexed when it contacts the ground, it could topple or become tilted in some way. Therefore, at the instant the leg touches the ground, if the leg is extended (by inhibiting flexion), then the leg will be able to support the weight of the robot.

We model this reflex as a simple inhibitory signal to the output function of the flexion ARR:

$$I_i = f_h\left(S_{fi} - \tau_i\right) \qquad (5)$$

where S_{fi} is a foot pressure in uncalibrated units, τ_i is selected to be above the noise threshold for the sensor. Compare this reflex to the extension and placement reflexes in cat (Kotliar et al. 1975).

2.3.2 Postural Control

Postural control is initiated by sensory feedback from the feet. The goal of the postural control module is to equalize pressure on the feet by making suitable shifts in trunk position and trunk configuration. We assume that the robot can measure pressure exerted on each foot. The input from the foot pressure sensors on the feet are compared and used to adjust bias of the posture. The postural adjustment is mediated by a low gain feedback from the sensors. The time constant is on the order of seconds. The equation for body posture adjustment is:

$$\tau \frac{d\vec{P}}{dt} = -\vec{P} + \Lambda \cdot \begin{bmatrix} A \\ B \\ C \end{bmatrix} \begin{bmatrix} S_{foot_1} \\ S_{foot_2} \\ S_{foot_3} \\ S_{foot_4} \end{bmatrix} \qquad (6)$$

where

$$A = \begin{bmatrix} 1 & -1 & -1 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}, C = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}, \Lambda = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda_2 & 0 \\ & 0 & \lambda_3 \end{bmatrix},$$

$$\vec{P} = \begin{bmatrix} L & Sw & Tw \end{bmatrix}^T$$

where λ_i are small constants controlling the adaptation rate. The term $P_1 = L$ controls the lunge of the robot, $P_2 = Sw$ controls the side-to-side sway, and $P_3 = Tw$ controls the body twist bias. The intuition behind the matrix mapping the sensor commands to bias commands is given by reference to Fig. 4. During walking, C=0. That is, twist compensation is switched off. This prevents the postural control mechanism from overriding the twist uCPGs.

2.4 Adaptive Modules (AM)

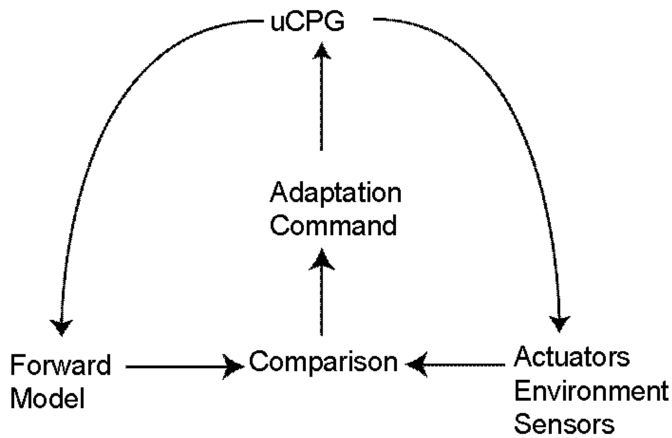


Figure 3. Use of innate internal model for CPG adaptation. The AM components are shown in the box composed of dotted lines.

In addition to the ARRs, another class of computational module, an adaptive module (AM), is responsible for altering parameters of the ARR. The basic model for an AM is shown in Fig.3. The AM has three components: (1) a *forward model* which uses an efference copy from a uCPG to predict sensory feedback, (2) a comparison of sensory feedback versus expected sensory feedback, and (3) a rule which uses the result of this comparison to modulate the uCPG in question.

Referring to figure 3, an ARR generates an output signal. This signal is sent to actuators and to a simple forward model. The purpose of the forward model is to form an expectation of a returning sensory signal. The actuator signal is propagated through the environment and eventually produces a sensory input to the robot. A comparison is made of the expected and the actual sensory signal. Based on the returning signal, the associated ARR is modulated to produce a different output. Below, we illustrate several AMs.

2.4.1 Twist Adaptive Module

This adaptive module acts with the ARR responsible for 'twist' movements of the body. A component of basic walking in animals is the sequential activation of the hypaxial muscles, responsible for twisting movements of the trunk in quadrupeds (Carrier 1990; Carrier 1993). Referring to Figure 4, the twist commands cause the trunk to rotate about its central axis. One of the effects of this motion is the transfer of weight to one set of diagonal feet and the unloading of the complementary diagonal set of feet.

The components of the Twist Adaptive Module are given below.

The forward model:

$$\frac{dm_{Tw+/-}}{dt} = -m_{Tw+/-} + y_{Tw+/-} \tag{7}$$

$$M_{Tw+/-} = f_h(m_{Tw+/-} - \tau_{tw}) \tag{8}$$

The comparison:

$$e_{Tw+} = f_s\left(S_{f1} + S_{f3}\right) \cdot M_{Tw+} \tag{9}$$

$$e_{Tw-} = f_s\left(S_{f2} + S_{f4}\right) \cdot M_{Tw-} \tag{10}$$

The correction rule:

$$A_{Tw+/-} = \int_0^T f_s\left(e_{Tw+/-}\right) - f_s\left(S_{v1} + S_{v2}\right) dt \tag{11}$$

$$A_{Tw+/-}(0) = 0 \tag{12}$$

where $y_{tw+/-}$ is the output of two ARR responsible for body twist. An output from x_{tw+} potentially causes feet 1 and 3 to become unloaded, and x_{tw-} potentially causes feet 2 and 4 to become unloaded. $S_{v1..2}$ are two vestibular signals indicating positive and negative roll, $S_{f1..4}$ are four foot pressure signals for feet 1..4. In addition $f_h(x)$ is a step function, $f_s(x) = \frac{1}{1 + e^{-\frac{(x-a)}{b}}}$, a sigmoid function.

The forward model works by filtering the twist command through a low pass filter. This introduces a slight time delay and thus simulates the propagation of the motor command to the robot's actuators, and the resulting sensory response. The resulting output of this filter is given as $m_e^{\pm}$ and thresholded to produce a binary expectation $M_{Tw+/-}$. When this variable is high, it indicates that no sensory input should be sensed. If $M_{Tw+/-}$ is active, and the appropriate pair of diagonally opposing feet sense pressure, then an error between the actual and desired condition is sensed.

The correction rule integrates the error. In practice, even when the system achieves a good behavior, there will be some residual error, as the forward model does not perfectly predict the sensory input. Even a small residual value will cause the twist amplitude to grow out of control. To check this, we introduce a term dependent on the vestibular system of the robot. As the twist increases, and the robot will begin a rolling motion and the vestibular system will become activated. This signal is used to suppress the amplitude. An equilibrium is found where the robot makes sufficient movement to unload its feet, yet small enough so that the body does not roll significantly.

2.4.2 Burst Length Adaptation for Leg Coordination

Another example of learning environmental coordination is burst length adaptation. As an animal walks, it must flex its legs to clear low-lying obstacles and minimize the possibility of stumbling or tripping. This flexion must terminate at precisely the correct moment. If termination is too soon, the leg will be fully extended while swinging forward and may strike the ground. If weight is transferred to the leg before it is extended enough, the robot will not have the correct geometry to remain stable, in the worst case, or the robot may walk in a semi-crouched, energy inefficient position, in the best case. A critical aspect of movement is to learn the precise duration of the flexion burst length (duration of burst length). The AM for burst length adaptation is given below.

Burst length forward model:

$$M_{KFi} = f_h\left(-\dot{x}_{KFi}\right) \tag{12}$$

The comparison:

$$e_{KFi} = f_b\left(S_{fi} + \dot{S}_{fi} - \tau_{bl}\right) \cdot M_{KFi} \tag{13}$$

The adaptive Rule:

$$\Delta BL = -e_{KFi} \cdot \beta \quad \text{for i=1,2,3,4} \tag{14}$$

$$BL(0) = \varepsilon \tag{15}$$

where $\dot{x}_{KFi}$ is the time derivative of burst of the ith knee flexor, and $f_b(x) = \begin{cases} -1 & x \le 0 \\ 1 & x > 0 \end{cases}$, the initial value of the burst length is a very small value, ε. If the burst is terminating, then M_{KFi} is active. That is, this forward model predicts that a sensory event occurs just as the burst is terminating. The comparison term examines a sensory signal and compares it to a threshold τ_{bl}. If there is no sensory information, when the burst terminates, this indicates that the burst length should increase. The term τ_{bl} insures this will happen. On the other hand, if the sensor stimulus is very high, either due to the foot being planted on the ground, or the foot force rapidly increasing, then the burst length was too long and should be shorted. The adaptive rule implements this.

2.4.3 Phase Adjustment

In a similar way, we can propose a rule to adjust the phase relationship between the hip and knee joints.

Forward model

$$M_{KFi} = f_h\left(\dot{x}_{KFi}\right) \tag{16}$$

Comparison

$$e_{kfi} = f_b\left(S_{fi} - \tau_{Phi}\right)\cdot M_{KFi} \tag{17}$$

Adaptive Rule

$$\Delta Ph = -e_{kfi} \cdot \beta_{Ph}\ M_{KFi} \tag{19}$$

$$Ph(0) = .Ph_0$$

As always, more complicated rules are possible (and have been successfully used) but for the sake of clarity, we choose a particularly simple rule. This rule finds a stable point on the slope of the decreasing sensor signal. The lower the τ_{phfi} parameter, the later the foot begins to flex. The beginning phase in the experiments below is selected to be 0.5. This starting value did not appear to be critical.

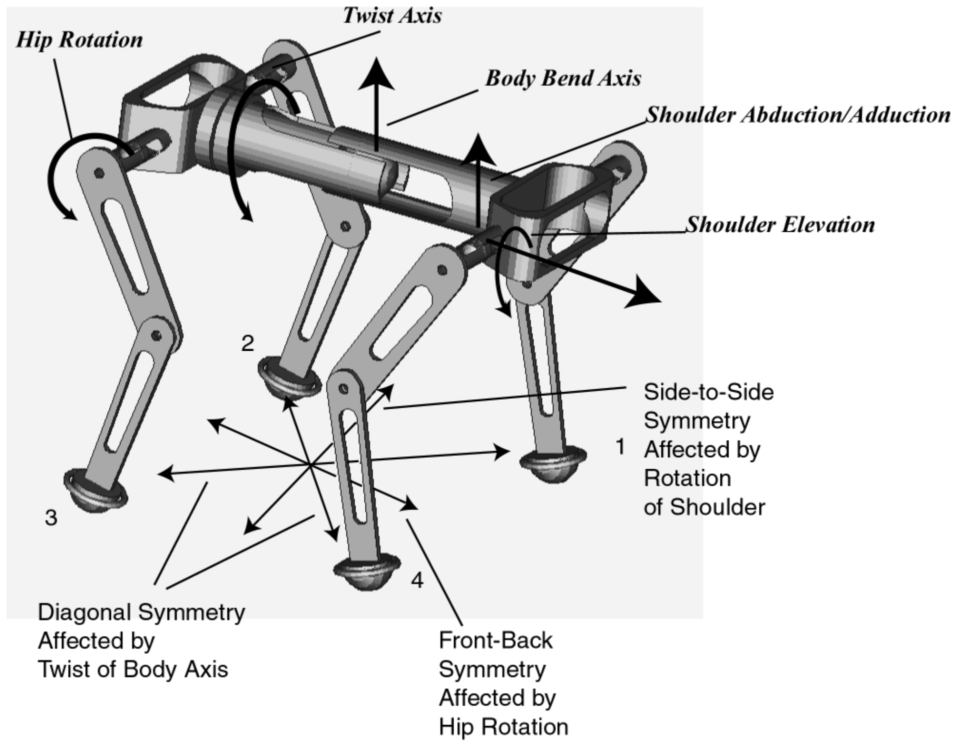


Figure 4. Postural Reflex Principle. Three types of symmetry are enforced for weight distribution. Diagonal— Comparison of diagonal feet; Front-to-Back— Comparison of front to back feet; and Side-to-Side— Comparison of feet on left to feet on right side. The numbers near the feet denote the numbering of the feet.

3.0 Experiments

The results of three experiments using GEO-II are described in this section. These experiments require progressively more adaptation to the environment and culminate in adaptive walking behavior. The robot learns to adjust key parameters of the CPG network to allow the robot to walk within minutes.

3.1 Experimental Setup

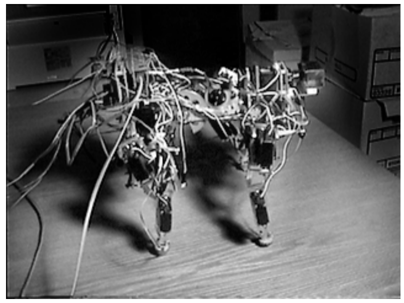


Figure 5. The GEO-II robot. GEO-II features a flexible spine.

The robot platform is a four-legged robot, "GEO-II" (Fig. 5). Sensor include a force sensor on each foot, and a gyro scope which senses body roll. The unique features of this robot include a flexible, three-degrees of freedom spine. This allows spinal movement including twist. A sketch of the basic structure of the trunk and legs is shown in Fig. 4. Model airplane servo actuators drive all axes. These servos are positional control devices. Geo II weighs 1.25 Kg.

Computation is divided between an onboard processor, a 68HC11 based ServoX24 board by Digital Designs and Systems, Inc., and a dual Intel Pentium workstation. The ServoX24 board is responsible for generating command signals for the servos as well as A/D sampling of sensor signals. The workstation is responsible for computing the ARRs, the AMs, and reflexes modules. The workstation also hosts a graphical user interface. All code runs under Windows 2000 (C++ Microsoft) in a multi-threaded, windowed environment.

3.2 Reflexes

Below, an experiment with the static postural reflex is reported. The extension reflex is not reported on as its behavior is straightforward. When the foot pressure sensor is touched, the leg extends regardless of the state of the uCPG driving the limb in question.

3.2.1 Static Postural Reflex

The result of the tuning should be to distribute weight equally over the feet of the robot. Initially, when the robot is placed on the ground, weight may not be symmetrically distributed. In other cases, disturbances caused by addition of payload, or other factors (e.g. a dragging cable) will disrupt weight distribution.

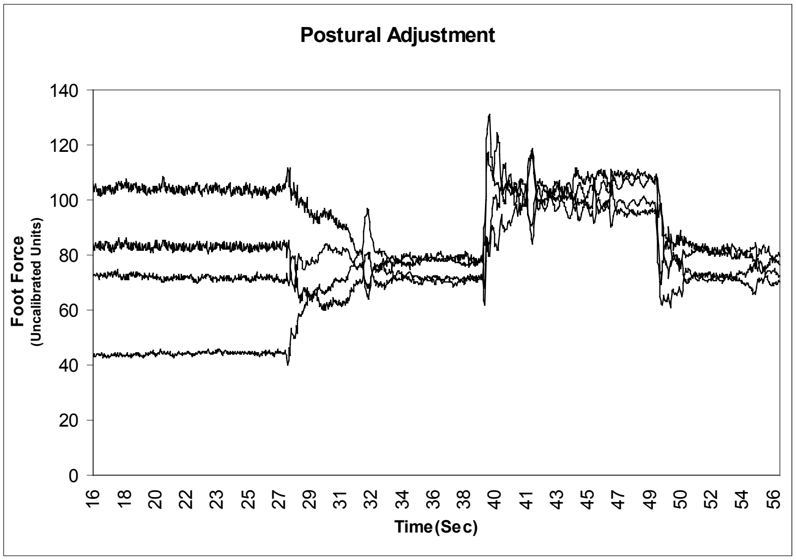


Figure 6. Postural Reflex Results. Graph of weight distribution among legs. Through time, the Posture AM gradually adjusts the weight distribution on the robot.

The robot was placed on the ground with the trunk in a random starting condition. Foot pressure recording begins. Referring to Figure 6, At t=16, the weight is not evenly distributed; the traces are not equal. At about t=29, the postural reflex is switch on. Through a period of several seconds, the posture is gradually adjusted, equalizing force on the feet.

Next a 360 gram payload (about 25% of the robot weight) is added over the front legs of the robot. At t=39, the payload is placed and then removed at about t=49 sec. At both the onset and the offset of the weight application, the robot makes a postural adjustment. When the weight is applied, the robot leans backward. When the weight is removed, the robot returns to its normal posture.

It was observed that if the robot is placed on a tilting platform, the robot adjusted its posture so that outstretched legs were vertical (parallel to the gravitational vector). Note that this was achieved by using foot pressure alone.

It was also observed that if the robot is pushed in any direction, the robot responds to this disturbance by resisting the push, creating an increase in force to counteract the disturbance.

3.3 Adaptive Modules

3.3.1 Twist Adaptation Results

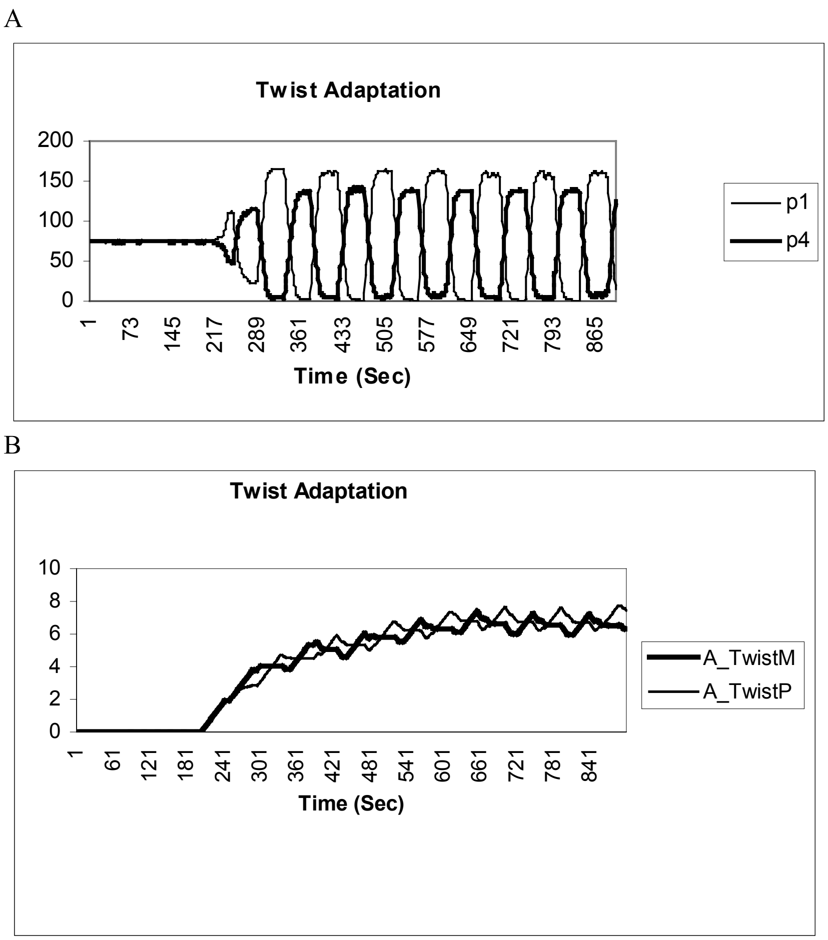
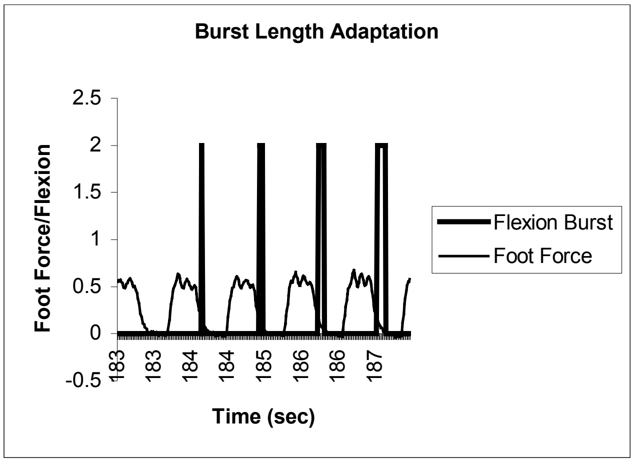


Figure 7. Twist Adaptation Response. (A) Foot pressure of front legs. (B) Change in amplitude A_i of the twist uCPGs.

The Twist ARR and Twist AM work together to generate a balanced, twisting of the body. When the Twist ARR is activated initially, the output gain of this module is near zero. Thus, no body twisting and no cyclic sensory feedback is sensed. The Twist AM gradually increases the gain as well as both the positive and negative twist uCPGs. (see Fig. 7b). Through time, as the robot shifts its weight, the paw forces vary periodically (see Fig. 7A). As can be seen, the front legs of the robot move out of phase. The magnitude of the twist amplitude is decreased due to gyroscope feedback.

In the end, the front legs periodically lifted symmetrically off the ground.

A



B

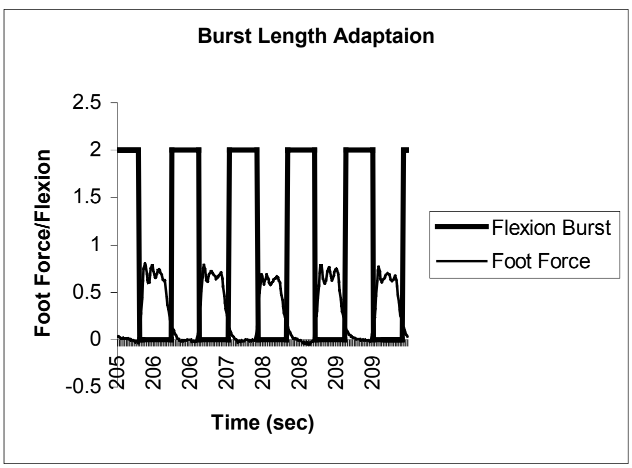


Figure 8. Flexion Burst Length Adaptation— Experimental Results. (A) Initially, the burst length is zero. Adaptation is turned on at 183 seconds. After adaptation is turned on, burst length increases. (B) Burst length has stabilized. As can be seen, burst length terminates just after ground contact.

3.3.2 Burst Length Adaptation Results

In this setup, GEO-II is suspended in a test apparatus. This is done to facilitate data collection only. If a trunk twist is induced, the hip assembly will rock back and forth and will cause the legs to contact the ground periodically. The experimental procedure is as follows: The hip is placed in oscillation, the burst length is reset to zero and adaptation is then turned on. Note that adaptation is independently controlled for each leg.

The adaptation of the left rear leg is shown in Fig. 8. Initially, the burst length is zero. As time progresses, the burst length lengthens and then stabilizes. As can be seen, the burst length adjusts until it stabilizes so that the burst terminates just as the leg touches the ground. If the leg were to terminate its flexion too early, this leg would be locked forward. This may cause the robot to stumble as it swings its foot forward. If it did not extend soon enough, the limb would be flexed when the leg struck the ground, and the robot may lose its balance.

3.4 Phase Adjustment and Burst Length Adjustment During Walking

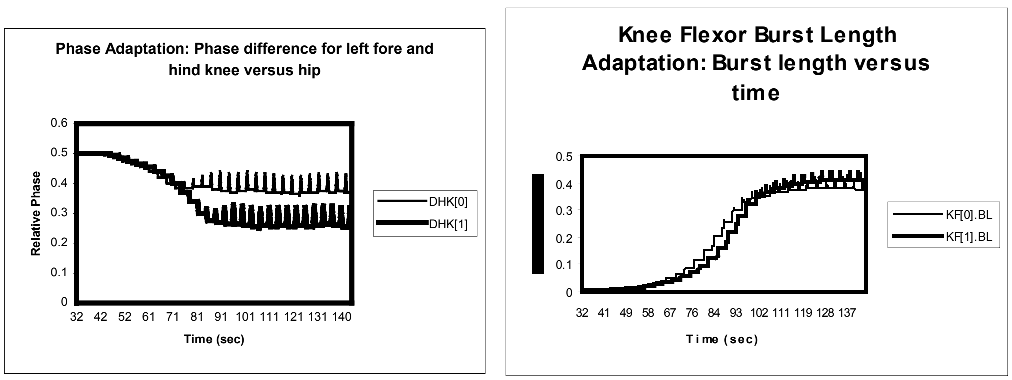


Figure 9. Phase and burst length adjustment in the walking robot. (A) phase adaptation during walking for the left front and hind legs. (B) Burst length adjustment during walking for the left front and hind legs.

We demonstrated adjustment of the phase of the knee movement versus hip. In this experiment, we select a nominal value for the relationship between the hip phase and leg phase of 0.5. Figure 9 (A) shows the phase adaptation through time for the left front and rear legs. As can be seen, the legs end with slightly different phase differences between the hip and knee.

Also shown in Fig 9 is the simultaneous adjustment of burst length during walking.

3.5 The Development of Walking

A walking gait can be achieved by applying the AMs in sequence or all at once. The hip flexion and knee flexion are given small fixed values. First, the postural reflex balances the robot, establishing a symmetric stance and facilitating the transition to walking. At this point, because the legs have not unweighted, the leg extension reflex prevents any leg from flexing. Further, because the robot is moving, the vestibular signal suppresses the twist module.

Next, the effect of the twist module begins to dominate. The robot shifts its weight back and forth to weight and un-weight the diagonal pairs of feet. As this effect increases, the weight on diagonal sets of feet decreases periodically (refer to Fig 7a).

During the time the leg is unweighted, the leg extension reflex is suppressed. The leg now has the potential to flex. The phase and burst length slowly adjust (see Fig 9A & 9B) from their initial values. After a minute or so, the legs begin to make quick flexion movements.

The flexion amplitude and hip swing amplitude are not controlled adaptively. Higher brain regions would probably control these variables. Hip swing amplitude controls stride length and leg flexion controls ground clearance during swing, and are goal dependent variables.

It should be noted that walking seemed to appear in *stages* although in this case, we did not explicitly introduced stages at which the adaptive modules were turned on. Instead, it seems that a natural sequence of apparent developmental behavior emerged spontaneously: First the robot balances, then it makes small movements of its trunk. This is followed by quick flexing movements of its limbs, and finally by a smoother walking movement. It is remarkable that these stages emerged spontaneously.

3.6 Robustness to changes in robot configuration

In this experiment, we added a 360 gram weight to the front of the robot. We repeated the walking experiment again. Again, the robot achieved a walking gait.

4.0 Discussion

Two key principles are combined in this work: (1) Innate forward models, (2) Distributed Control.

Using these principles, the following three problems are addressed: (1) Static Postural Control, (2) Learning proper parameter settings for the output of a uCPG model (an ARR), and (3) Learning coordination with the environment.

Postural control is of primary importance. The postural control mechanism in this work relies on force sensors on the foot as the sole input to the control system. The idea behind the control scheme is to maintain three degrees of symmetry— front to back, side-to-side and diagonal symmetry. Thus sensory signals become references for other signals related by a degree of symmetry; senses are compared to each other. Furthermore, each degree of symmetry is related to a particular 'muscle group' in the robot. Front to back symmetry generates commands for the hip rotation axes. Side-to-side symmetry drives hip adductors. Finally, diagonal symmetry drives twist about the body axis. It is fortunate that these degrees of symmetry map so easily onto the proximal actuators. In other mechanical walking machines, the twist axis is fixed. Thus asymmetries must be compensated for by leg flexion. By adding the twist, the torso and associated proximal actuators can compensate for imbalances (as defined here). It is also noted that compensation could be achieved on a slope. Also, postural adjustment can be made to compensate for weights placed on the robot. Such weight may mimic a payload dropped onto the robot's back. It may also mimic constant perturbations such as dangling wires and cables. Thus the system, by using the principle of symmetry, can simply drive muscles in a one-to-one fashion without the need for any detailed kinematic model of the robot. Yet this scheme achieves postural control in the face of significant outside disturbances.

The issue of control from higher level centers is not discussed here. The general strategy is to allow direct parameter modification from higher centers. The interested reader is directed to (Lewis and Simó 1999; Lewis and Simó 2001) for an example of using high level vision to adaptively control ARR. In that work, an example is given of Burst Length Modulation by a higher center (communicating with visual centers).

Adaptive modules are used to tune the output of a CPG. Here the idea is to use a model of sensory expectation. The models are rudimentary and are sufficient to allow the system to bootstrap. It is not necessary to have any detailed model of the dynamics of the system.

Adaptive modules were also used to generate expectancies for when a foot strike should occur. This allows the adjustment of the flexor burst length. This adaptive mechanism was crucially important in assisting the robot in generating precisely the correct burst timing, preventing stumbling and foot dragging. Again the model used was rudimentary and did not depend on any detailed dynamics of the robot.

This idea of a rudimentary model is crucially important to understanding the value of this work. The idea of a detailed dynamic model is rejected. The power in this idea is that the adaptive system should, in principle, work over a very broad range of robotic devices with similar form. We made only the most general assumptions about the structure being controlled here. This also means that it is not necessary to identify a model (i.e. to instantiate by making detailed perturbations and observations) before learning can begin. That is why learning was achieved so rapidly in the cases presented.

The results of the present work illustrate the power of simple innate models in bootstrapping the system. In earlier work, it was assumed that adaptation should be made to occur in stages (Lewis et al. 1992). However, remarkably, all adaptive modules could be switched on simultaneously, and developmental stages seemed to emerge spontaneously. This was surprising.

The author suggests that once the system has overcome the basic problem of developing a rudimentary gait, using the techniques defined here, more general-purpose learning would continue to refine the walking over time. In vertebrates, this may be one of the roles of the cerebellum.

Once the robot has obtained a rudimentary gait, it is necessary that this gait become more optimal with time. Here, optimality might be related to minimize the wear and tear on joints or minimizing visual disturbances. Work in that area of movement optimization will be reported in the future.

5.0 Summary and Conclusions

CPGs have been well studied by a number of researchers with possible applications to the control of walking machines. While a simple CPG circuit, consisting of a handful of oscillators, can be constructed, it is not clear how these CPGs can be made to adapt to a particular machine.

The approach presented here has the following elements: (i) A CPG model is chosen where the parameters controlling the behavior of the model are represented explicitly. This model is called an Adaptive Ring Rule (ARR). (ii) Included here is the idea of an innate internal model. That is, a model that predicts, even in a crude way, the sensory consequences of intended action. These models are used by Adaptive Modules (AMs) to alter the parameters of the ARR so that the sensory feedback more closely resembles the output of associated innate internal models.

By choosing the correct AMs, we demonstrate that critical elements of gait, such as flexor burst length adaptation, hip-knee phase, and the twisting of the body, can quickly be acquired. These AM and ARR, in conjunction with two basic reflexes (postural control and foot extension reflex) allow the robot to quickly acquire a basic trot gait within minutes of inception. This quick learning is compatible with the learning exhibited by animals such as horses immediately after birth. Further, such adaptive mechanism can be built into custom electronic circuits, which are under development (Lewis et al. 2000; Lewis et al. 2001)

Acknowledgments

This work was supported in part by ONR N0014-02-C0110, ONR N0014-99-1-0984, N00014-96-10657, as well as a grant from the Beckman Institute of the University of Illinois. The author wishes to thank Lucia Simo, Mitra Hartmann and Nicolas Schweighofer, and Chis Milne for valuable feedback on the manuscript. The author is also grateful for feedback from anonymous reviewers.

References

Bay, J. S. and H. Hemami (1987). "Modeling of a neural pattern generator with coupled nonlinear oscillators." IEEE Trans. on Biomedical Engineering, **BME-34**(4): 297-306.
Bekoff, A. (1985). Development of Locomotion in Vertebrates, a Comparative Perspective. The comparative Development of Adaptive Skills: Evolutionary Implications. E. S. Gallin. London, Lawrence Erlbaum Assoc.**:** 57-94.
Carrier, D. (1990). "Activity of the hypaxial muscles during walking in the lizard *Iguana Iguana*." J. Exp. Biol **152**: 453-470.
Carrier, D. R. (1993). "Action of the hypaxial muscles during walking and swimming in the salamander *Dicamptodon Ensatus*." J. Exp. Biol **180**: 75-83.
Cohen, A. H. (1988). Evolution of the vertebrate central pattern generator for locomotion. Neural control of rhythmic movements in vertebrates. S. R. A. H. Cohen, S. Grillner. New York, Wiley.
Cohen, A. H., P. J. Holmes, et al. (1982). "The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical model." J. Math. Biol. **13**: 345-369.
Collins, J. J. and I. N. Stewart (1993). "Coupled Nonlinear Oscillators and the Symmetries of Animal Gaits." J. Nonlinear Sci **3**: 349-392.
Grillner, S. and P. Wallén (1985). "Central pattern generators for locomotion, with special reference to vertebrates." Annual Review Of Neuroscience **8**: 233-61.
Grillner, S. and P. Zangger (1975). "How detailed is the central pattern generation for locomotion?" Brain Research **88**(2): 367-71.
Ilg, W. and K. Berns (1995). "A Learning Architecture Based on Reinforcement Learning for Adaptive Control of the Walking Machine LAURON." Robotics and Autonomous Systems **15**: 321-334.
Ito, S., H. Yuasa, et al. (1998). "A mathematical model of adaptive behavior in quadruped locomotion." Bio Cybern **78**: 337-347.
Jalics, L., H. Hemami, et al. (1997). "A control strategy for Terrain Adaptive Bipedal Locomotion." Autonomous Robots **4**(3): 243-257.
Kawato, M. and D. Wolpert (1998). "Internal models for motor control." Novartis Found Symp **218**: 291-304; discussion 304-7.
Kimura, H., S. Akiyama, et al. (1999). "Realization of Dynamic Walking and Running of the Quadruped Using Neural Oscillator." Autonomous Robots **7**(3).
Kotliar, B. I., V. I. Maorov, et al. (1975). "Models of learning based on the plastic properties of the "placing
reaction" in cats." Zh Vyssh Nerv Deiat **25**(5): 967-973.
Lewis, M. A. (1996). Self-organization of Locomotory Controllers in Robots and Animals, Ph.D. Dissertation. Department of Electrical Engineering. Los Angeles, University of Southern California.
Lewis, M. A., R. Etienne-Cummings, et al. (2000). Toward Biomorphic Control using Custom aVLSI Chips. 2000 International Conference on Robotics and Automation, San Francisco, IEEE.
Lewis, M. A., A. H. Fagg, et al. (1992). Genetic Programming Approach to the construction of a

neural network for control of a walking robot. 1992 IEEE International Conference on Robotics and Automation, Nice, France.

Lewis, M. A., M. Hartmann, et al. (2001). "Control of a Robot Leg with an Adaptive aVLSI CPG Chip." Neurocomputing (Procedings of the Computational Neuroscience Meeting) **38-40**: 1409-1421.

Lewis, M. A. and L. S. Simó (1999). "Elegant stepping: A Model of Visually Triggered Gait Adaptation." Connection Science **11**(3/4).

Lewis, M. A. and L. S. Simó (2001). "Certain Principles of Biomorphic Robots." Autonomous Robots **11**(3): 221-226.

Matsuoka, K. (1987). "Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators." Biol. Cybern **56**: 345-353.

Murray, J. D. (1993). Mathematical Biology. Berlin, Springer-Verlag.

Rand, R. H., A. H. Cohen, et al. (1988). Systems of Coupled Oscillators as Models of Central Pattern Generators. Neural Control of Rhythmic Movement in Vertebrates. A. H. Cohen, S. Rossignol and S. Grillner. New York, Wiley.

Sperry, R. W. (1950). "Neural basis of the spontaneous optokinetic response produced by visual inversion." Journal of Comparative and Physiological Psychology **43**: 482-489.

Taga, G., Y. Yamaguchi, et al. (1991). "Self-Organized Control of Bipedal Locomotion by Neural Oscillators in Unpredictable Environment." Biol. Cybern. **65**: 147-159.

Zielinska, T. (1996). "Coupled oscillators utilised as gait rhythm generators." Biol. Cybern. **74**: 263-273.