ELSEVIER

Robotics and Autonomous Systems 16 (1995) 135-159

Robotics and Autonomous Systems

A comparison of three insect-inspired locomotion controllers

Cynthia Ferrell [1,2]

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 545 Technology Square, Room 741, Cambridge, MA 02139, USA

Abstract

This paper compares three insect inspired controllers which were implemented on an autonomous hexapod robot. There is a growing interest in using insect locomotion schemes to control walking robots. Researchers' interest in insect-based controllers ranges from understanding the biological basis of locomotion control in insects to building real-time walking machines which require relatively little computational power. Several models for insect locomotion exist, and robotics researchers tend to adopt one approach and experiment with it.

In contrast, this paper offers a comparison of three insect inspired controllers - all of which were implemented and tested on the same autonomous hexapod robot. Some of the controllers used reflex-based mechanisms whereas others used pattern-based mechanisms. Reflexive controllers exploit sensory stimulus and response reactions to produce leg motion and gait coordination. In contrast, pattern-based controllers depend more upon pre-programmed patterns of behavior which may be influenced by external events. Typically, these pre-programmed patterns of behavior are implemented using central pattern generators (CPGs).

In this work, we compare gait coordination performance of three controllers on flat terrain. We extend the comparison to include leg loading considerations, disabled leg compensation, and externally applied leg perturbations. We discuss the differences between controllers with respect to inconsistent leg retraction velocities, leg design issues, sensing requirements, and computational issues. The robot performed quite differently under varying experimental conditions depending upon which controller was used. We found that controller performance was the most sensitive to robot design parameters. For our case, we had the most success with pattern-based mechanisms given the leg design of our robot and its limitations in controlling the retraction velocity of its legs. The pattern-based mechanisms allowed the robot to remain stable over a variety of gaits while the robot was subjected to loading the legs, disabling a leg, and physically disturbing the legs. The reflexive mechanisms were less successful at maintaining stability when the robot's legs were increasingly disrupted.

1. Introduction

There is a growing number of researchers investigating biologically motivated models of insect locomotion. Interest in these areas is quite varied, and there is far more work on this topic than is mentioned here. Some researchers are interested in determining the biological basis of arthropod locomotion. For example the authors of Refs. [14,27] have proposed models and mechanisms to explain locomotion for a variety of species. Other researchers, such as Dean [18] explore the validity of such models in simulation, while others have used hexapod robots [4,30,17]. A number of researchers are in pursuit of better walking

[1] Support for this research was provided in part by a NASA Graduate Student Researcher Program Fellowship administered through the Jet Propulsion Laboratory, by Jet Propulsion Laboratory grant 959333, and by the Advanced Research Projects Agency under Office of Naval Research contract N00014-91-J-4038.

[2] Tel: (617)253-0997; Fax: (617)253-0039; E-mail: ferrell-@ai.mit.edu

SSDI 0921-8890(95)00147-6

robots, and a variety of work demonstrates the appeal of implementing insect-inspired schemes on robots. Insect motivated approaches have been used to produce real-time locomotion with limited computational power [19,7]. This is particularly important for small autonomous hexapod robots that carry their own computation [1,20]. Refs. [7,20,17] have shown that biologically motivated schemes exhibit good responsiveness to gait perturbations caused by the environment (i.e. holes or objects in the robot's path). Refs. [9,20] have also demonstrated the robustness of insect-like controllers to various controller lesions.

There is a growing number of researchers using insect-like controllers to control legged robots, but there is little work comparing the various methods. In this work, we investigate three insect inspired controllers. To mimic insect locomotion, locomotion control is distributed among the six legs. Each local leg controller is responsible for generating the cyclic motion of its leg, and the local controllers influence each other to synchronize and coordinate leg behavior. Each controller was implemented and evaluated on our hexapod robot. Some of the controllers use more reflex-based mechanisms while others use more pattern-based mechanisms to control locomotion. Reflexive controllers exploit sensory stimulus and motor response reactions to produce leg motion and gait coordination. Pattern-based controllers depend more upon pre-programmed patterns of behavior which can be influenced by sensed external events. Typically these pre-programmed patterns of behavior are implemented with central pattern generators (CPGs). Most biologically motivated controllers exhibit one or both kinds of mechanisms to varying degrees. However, despite the different nature of these controllers, each produces characteristics of insect locomotion as described by Ref. [31].

This paper is organized as follows. First we describe the robot used in our experiments. Next, we present the biological basis of each controller. The controllers presented were either implemented from proposed models of insect locomotion, or pieced together using biologically motivated mechanisms. Given this background, we present some related work involving insect-like controllers on hexapod robots. We then introduce the implementation of the three insect-inspired controllers on our robot. We compare the gait generation performance of each controller while the legs are unloaded (walking while suspended above ground) and loaded (walking on flat terrain). Unfortunately, the insect models we investigated only account for flat terrain locomotion. Hence, we modified the flat terrain controllers so the robot could accommodate disabled legs or externally applied leg disturbances. Not all of these modifications were biologically motivated, but it gave us an opportunity to see how readily each controller supports adding new behaviors. This reflects our bias of using insect locomotion models to enhance robot locomotion, as opposed to using robots to study insect locomotion. The controllers are evaluated with respect to gait stability, gait variety, and efficient leg coordination. We found the robot performed quite differently depending upon which controller was used. We compare and discuss the performance differences with respect to leg retraction velocity control, leg design issues, sensing capabilities, and computational requirements.

2. Experimental platform

Hannibal, as shown in Fig. 1, is a small autonomous robot. It was designed and built under the supervision of Prof. Rodney Brooks in the Artificial Intelligence Lab at MIT [1]. The robot is relatively sophisticated and complex robot for its size. It measures 35 cm long, stands 15 cm high, and weighs 2.8 kg. It has six 3 degree of freedom (DOF) legs (see Fig. 2). Despite its small size, Hannibal currently has 19 actuators and over 60 sensors of 5 different types all connected via a local network to 8 on board computers. The robot's control organization is behavior-based [6]. It is programmed in the Behavior Language, a front end for the Subsumption Architecture [8].

The following types of sensors are mounted on Hannibal:

• Leg mounted force sensors: these are foil strain gauges that can be used to measure loads on the leg and to detect leg collisions. There is a set of strain gauges for each DOF of the legs.

• Joint angle sensors: These are potentiometers that measure the joint angle for each DOF of the leg.

• Joint velocity sensors: The joint angle sensors are differentiated in analog for each DOF of the leg.

• Ground contact sensor: This is a linear potentiometer mounted on the ankle that measures the de-

Fig. 1. Hannibal is quite complex for its size. It is approximately the size of a bread box and is equipped with 19 degrees of freedom, over 60 sensors, and 8 computers.

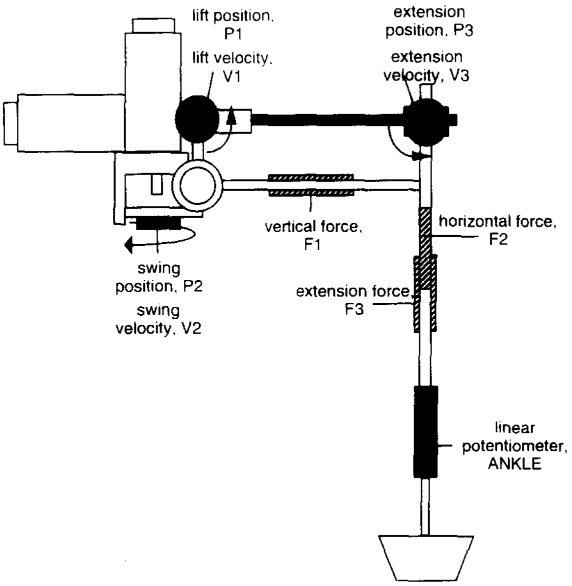


Fig. 2. Schematic of Hannibal's leg. Hannibal has six legs, each with three degrees of freedom (DOF). The swing DOF advances and retracts the leg by rotating it about the shoulder (P2), the lift DOF raises and lowers the foot by rotating the upper link about the shoulder (P1), and the extension DOF lifts the foot by extending the lower link about the elbow (P3). All six legs are identical.

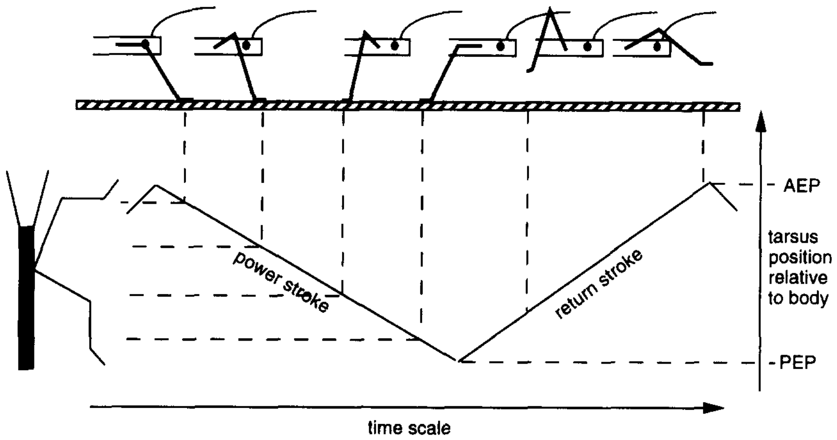


Fig. 3. During the recovery phase, the leg lifts and swings to the start position of the next power stroke. In forward locomotion, the leg moves towards the AEP during the recovery phase. During the support phase, the leg supports and propels the body along the direction of motion. In forward walking, the leg moves toward the PEP during the support phase. Adapted from [16].

flection of the foot as it presses against the ground.

• Inclinometer: This sensing unit is made up of a ± 45 degree roll sensor and a 360 degree pitch sensor.

3. Insect locomotion

Insect locomotion is exceptionally robust, adaptive, and versatile. We would like Hannibal to walk with the same qualities. Toward this goal, we have investigated three models of insect locomotion. The first model presented is a descriptive model proposed by [31]. Although this model does not propose specific mechanisms, it nicely summarizes some of the primary characteristics typical of insect locomotion. Next, we introduce a model proposed in [16]. This model is strongly reflexive in both individual leg control as well as in gait coordination. Finally, a model suggested by Pearson, as described in Ref. [31], is presented. In contrast to Ref. [16], Pearson's model is strongly patterned with respect to individual leg control and gait coordination. Not surprisingly, some scientists have proposed a combination of these two paradigms.

3.1. Definition of terms

Below are several terms we use through out this paper. (See Fig. 3.)

(1) *Protraction*: The leg moves towards the front of the body.

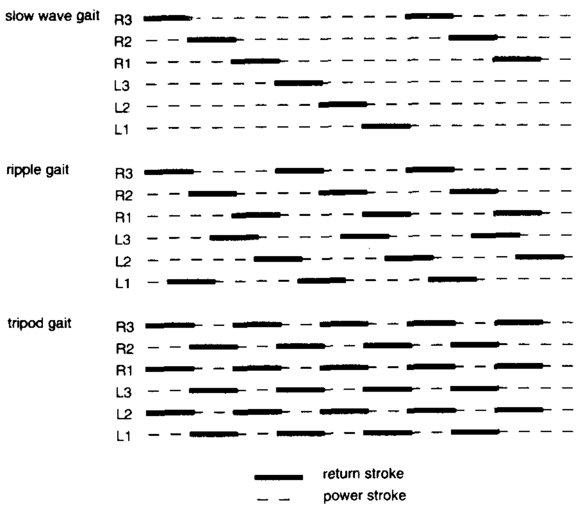


Fig. 4. Some commonly observed gaits of insects. All are members of the family of wave gaits. Adapted from Ref. [31].

(2) *Retraction*: The leg moves towards the rear of the body.

(3) *Power stroke*: The leg is on the ground where it supports and propels the body. In forward walking, the leg retracts during this phase. Also called the *stance phase* or the *support phase.*

(4) *Return stroke*: The leg lifts and swings to the starting position of the next power stroke. In forward walking, the legs protracts during this phase. Also called the *swing phase* or the *recovery phase.*

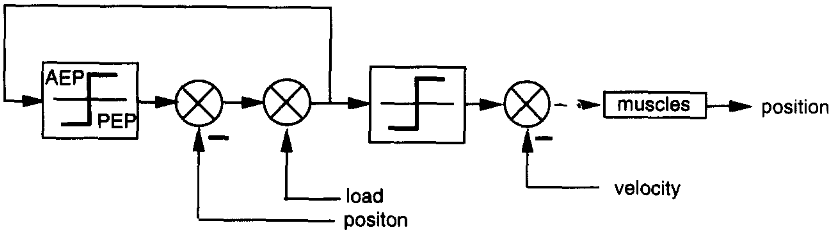


Fig. 5. Cruse's circuit for controlling the motion of an individual leg. The left side of the circuit determines whether the system adopts the power stroke or the return stroke. The left relay characteristic produces the two alternative target positions, AEP or PEP, when its input value is positive or negative, respectively. The value of the target position is compared with the actual leg position, which is then sent as a reference input to the right side of the circuit. The right side of the circuit is a velocity-controlling feedback system that causes the leg to move to the target position according to the reference input. Adapted from [16].

(5) *Anterior extreme position (AEP)*: In forward walking, this is the target position of the advance degree of freedom during the return stroke.
(6) *Posterior extreme position (PEP)*: In forward walking, this is the target position of the swing degree of freedom during the power stroke.

3.2. Wilson

Ref. [31] presents a descriptive model for characterizing all of the commonly observed gaits of cockroaches, including those resulting from amputation. Some of these gaits are shown in Fig. 4. These rules describe most of the qualitative features of leg coordination in cockroaches when they walk on smooth horizontal surfaces, however they do not account for all the empirical data.

- A wave of protraction runs from posterior to anterior. No leg protracts until the one behind is placed in a supporting position.
- Contra-lateral legs of the same segment alternate in phase.
- Protraction time is constant.
- Frequency varies (retraction time decreases as frequency increases).
- The intervals between steps of the hind leg and middle leg and between middle leg and foreleg are constant, while the interval between the foreleg and hind leg steps varies inversely with frequency.

3.3. Cruse

Holk Cruse has studied locomotion of several animals [11-13,15,16]. Among other models, he developed two models for the locomotion of walking stick insects *Carausius morosus*. The first is a model for the control of individual legs; the second is a model for the coordination between legs.

Cruse's model for the control of an individual leg is presented in Ref. [14] and is shown in Fig. 5. Reflexive mechanisms in each leg generate the step cycle of the leg by evoking transitions between the power stroke and the return stroke. Each leg transitions from the return stroke to the power stroke when the leg reaches the AEP. Each leg transitions from the power stroke to the return stroke when the leg reaches the PEP, the load on the leg is small, and the adjacent legs are in their supporting phase. These conditions insure the leg does not lift until the body is supported by the other legs.

In Ref. [15], the author presents six mechanisms that are responsible for the coordination between legs. These mechanisms are redundant in re-establishing coordination in the case of minor disturbances. We present only the three primary mechanisms. These mechanisms affect the threshold for beginning the return stroke by adjusting the PEP of the receiving leg. The PEP adjustment is based on the sum of the interleg influences affecting that leg. The threshold for beginning the power stroke (AEP) is fixed. The influences are sent between legs as shown in Fig. 6. Fig. 7 shows what these influences look like as a function of leg position.

- *Mechanism*$_1$: Rostrally directed influence inhibits the start of the return stroke in the anterior leg by shifting the PEP to a more posterior position. This is active during the return stroke of the posterior leg.
- *Mechanism*$_2$: Rostrally directed influence excites the start of the return stroke in the anterior leg by shift-

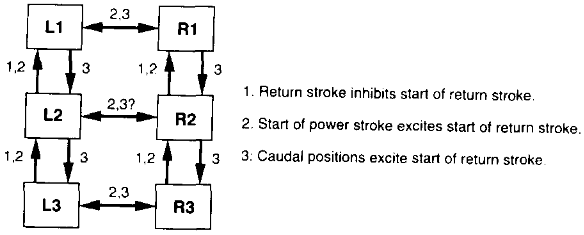


Fig. 6. Cruse's circuit for leg coordination. This figure summarizes the coordinating mechanisms operating between the legs of a stick insect. Each leg is represented by a box in the diagram. L1 and R1 correspond to the left front and right front legs, L2 and R2 correspond to the left middle and right middle legs, and L3 and R3 correspond to the left rear and right rear legs, respectively. The numbers above the arrows represent which influences are sent between legs. The direction of the arrows represent which way the influences are sent between the legs. Adapted from Ref. [16].

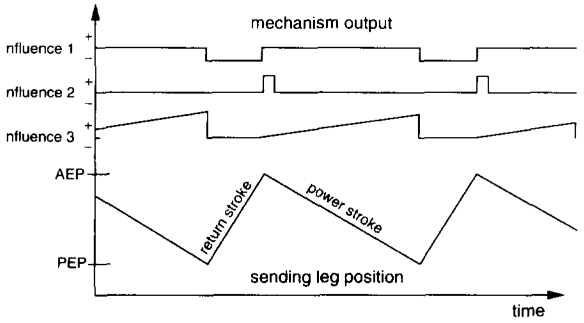


Fig. 7. A diagram showing the relationship between leg position of the sending leg and the coordinating influences it sends. An excitatory influence corresponds to positive values, and an inhibitory influence corresponds to negative values. This figure illustrates the time and duration of the coordinating influences as a function of leg position.

ing the PEP to a more anterior position. This is active during the start of the power stroke of the anterior leg. It also applies to the contra-lateral leg.

• *Mechanism*$_3$: Caudally directed influence excites the start of a return stroke in the posterior leg. The start of the return stroke is more strongly excited (occurs earlier) as the anterior leg is moved more rearward during the power stroke. This causes the posterior leg to perform the return stroke before the anterior leg begins its return stroke. This is active during the power stroke of the anterior leg. It also applies to the contra-lateral leg.

How do these three mechanisms stimulate stability and back to front metachronal waves? As an example, let us discuss their influence on the right front leg, R1. For this discussion we use the leg labeling convention of Fig. 6. We say *mechanism*$_1$ exerts *influence*$_1$, *mechanism*$_2$ exerts *influence*$_2$, and *mechanism*$_3$ exerts *influence*$_3$. R1 receives four influences: *influence*$_2$ and *influence*$_3$ from L1 and *influence*$_1$ and *influence*$_2$ from R2. *Influence*$_1$ from R2 inhibits R1 from beginning its swing phase while R2 is in its swing phase. This encourages stability by discouraging R1 from lifting until R2 is supporting the body. *Influence*$_3$ from L1 stimulates R1 to start its swing phase while L1 is in support phase. This influence strengthens as L1 approaches its PEP so that R1 will perform its return stroke before L1 finishes its power stroke. This also encourages stability. *Influence*$_2$ from R2 and L1 encourages R1 to begin its return stroke just after either R2 or L1 begin their power stroke. This establishes back to front metachronal waves for adjacent ipsilateral legs and 180 degree phasing between contra-lateral legs. R1 sends three influences: *influence*$_2$ and *influence*$_3$ to L1 and *influence*$_3$ to R2. Sending *influence*$_3$ to L1 and R2 contributes to stability by exciting L1 and R2 to swing while R1 supports the body. Sending *influence*$_2$ to L1 encourages 180 degree phasing between contralateral legs of the same segment.

3.4. Pearson

Keir Pearson and his collaborators investigated the neural systems that control walking in the cockroach [31,27]. They developed neurological models to explain the control of an individual leg and the coordination between legs.

In their model, the control of an individual leg is pattern-based. The complex unit of action that controls the stepping pattern of a single leg combines three elementary units of action – the oscillator, the servomechanism, and the reflex (see Fig. 8). The oscillator generates the stepping pattern of the leg by controlling the activation of the flexor motor neurons and the extensor motor neurons. The effectors move as a function of oscillator phase, hence they behave in a pre-programmed manner. The flexor motor neurons protract the leg when activated, and the extensor motor neurons retract the leg when activated. At the peak of its cycle, the oscillator generates the leg's swing phase by activating the flexor motor neurons and inhibiting the extensor motor neurons. The duration of this protraction command is independent of the oscillator's

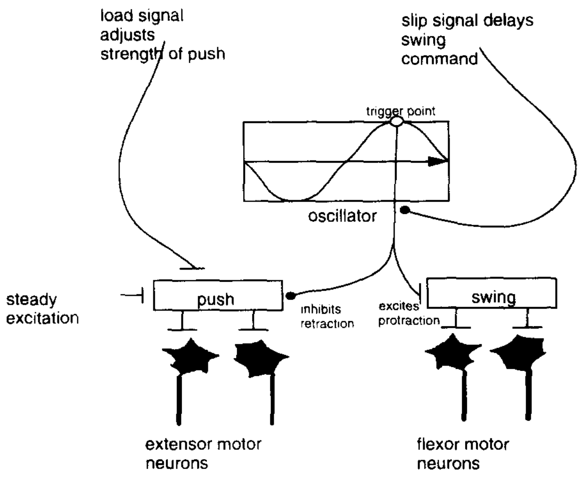


Fig. 8. Pearson's neural circuit for controlling the stepping motion of a single leg. An oscillator provides the stepping rhythm. It triggers a swing command near the peak of its cycle. The swing command excites the motor-neuron circuit that swings the leg forward, and inhibits the push circuit. The push circuit presses the foot to the ground and draws it back. A steady excitatory input keeps the push circuit active whenever it is not inhibited by the swing command. Adapted from Ref. [31].

period, so the protraction time is constant. Throughout the remainder of the oscillator's cycle, the oscillator generates support phase by activating the extensor motor neuron. The retraction rate varies with the frequency of the oscillator. The rhythmic pattern established by the oscillator is modified by a servomechanism circuit and a reflex circuit. The servomechanism circuit uses sensory signals fed back to the central nervous system from joint receptors and/or stretch receptors. The sensory feedback adjusts the strength of the supporting and pushing contractions to match variations in load. The reflex circuits delay or prevent the command that swings the leg forward. The input to these reflexes comes from receptors that detect whether another leg has taken up some of the load. For example, if a middle leg is amputated, it fails to take up the load at the normal time. This failure delays the protraction of the front leg. When the rear leg hits the ground it takes up some of the load, and this releases the delayed protraction of the front leg. As a result the front and hind legs step 180 degrees out of phase instead of in phase (transitions from a tripod gait to a slower wave gait).

Gait generation is composed of the pre-programmed patterns of behavior described above. The unit that controls walking is comprised of six leg-stepping units, one for each leg, and a command neuron(s). These leg-stepping units are coordinated by coupling signals that pass back and forth between the oscillators. The oscillators that control contra-lateral legs of the same segment maintain a constant 180 degrees phase relationship. Hence, contra-lateral legs of the same segment alternate in phase. In contrast, the three oscillators along either side of the body maintain the same temporal lag. The fixed lag between ipsilateral oscillators keeps the stepping intervals constant between the hind leg and middle leg, and between the middle leg and the front leg. The fact that the front oscillator lags the middle, which in turn lags the rear oscillator, means that a wave of protractions runs from posterior to anterior. The final component of the walking circuit is the command neuron(s), which sets the pace of walking by changing the period of the oscillators. A strong command signal makes the oscillators cycle rapidly (decreases the oscillator period). Oscillators with a fixed-lag coupling must change their phase relationship when their period changes. Consequently, the changes in gait are simply changes in the phase relationship between the three oscillators on either side. See Fig. 9.

4. Related work

Several researchers have implemented insect-like controllers in simulation or on robots. These controllers span the reflexive/patterned space – some are purely reflexive, some are purely patterned, and some are a little of each. We briefly present some representative controllers in this section, and compare them to our implementations in Section 7 of this paper.

4.1. Reflexive controllers

Refs. [4,17] have implemented locomotion controllers based on the mechanisms presented in Section 3.3.

Ref. [4] successfully tested the three mechanisms proposed by Cruse (and described in Section 3.3) on a small hexapod robot with 2 DOF legs. By removing various mechanisms, Beer and his colleagues determined that *mechanism*$_2$ promotes normal back to front metachronal waves, and *mechanism*$_3$ promotes 180

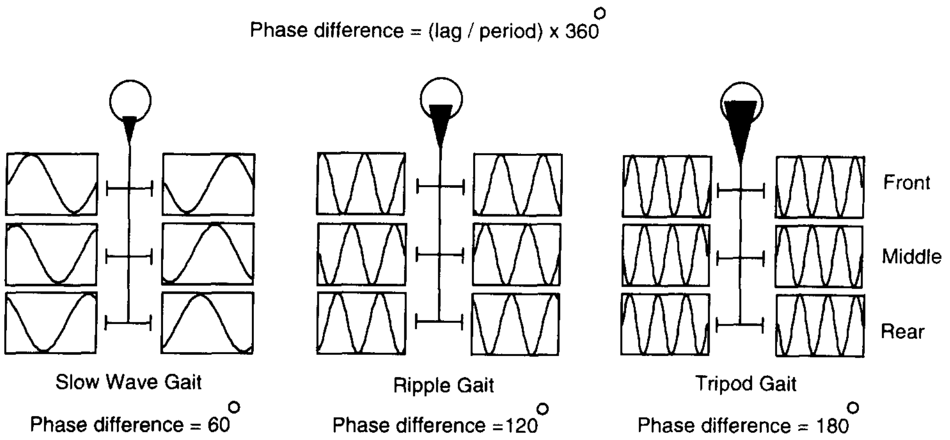


Fig. 9. Various gaits emerge from changing the frequency of the local leg oscillators. Changing the phase difference between ipsilaterally adjacent legs produces different gaits. As shown in the figure, a phase difference of 60 degrees produces a slow wave gait, a phase difference of 120 degrees produces a ripple gait, and a phase difference of 180 degrees produces a tripod gait. The phase difference between contra-lateral legs of the same segment is fixed at 180 degrees.

degree phasing between cross-body leg pairs. They found *mechanism*$_3$ is the most effective single coordinating mechanism and *mechanism*$_2$ was the least effective. They report that these mechanisms produce stable metachronal coordination over a wide range of step periods.

Ref. [17] describes a neural net implementation of his insect model. Although Ref. [17] only presents results in simulation, a similar controller was implemented on a large hexapod robot with 3 DOF legs [30]. The simulated controller is quite robust. It produces appropriate return stroke movements over a variety of initial and final positions, rapidly reestablishes a stable gait after disturbing leg movement, and retracts in response to collisions during the swing phase. The robot has demonstrated similar performance.

4.2. Hybrid controllers

The authors of Refs. [19,7,2] have implemented hybrid controllers on their robots.

In the 1980's, Marc Donner implemented a distributed locomotion controller on the SSA Hexapod [19]. The SSA hexapod is much larger than Hannibal. It had six 3 DOF legs which were hydraulically actuated, and it had position and force sensing for each DOF (among other sensors). Donner's walking algorithm was inspired by insect locomotion. Individual leg control was patterned in nature; it was implemented by a separate and mostly autonomous process responsible for generating preprogrammed cyclic stepping trajectories. Leg coordination was reflexive; it was implemented by an excitation mechanism and an inhibition mechanism. Each leg process sends/receives these influences to/from neighboring leg processes. The excitation and inhibition influences determine when a leg makes the transition from the support phase to the swing phase. Using this approach, the SSA hexapod walked as well as the physical constraints of the machine allowed. The approach also permitted the machine to locomote when a middle leg was removed.

Genghis is a small hexapod robot (35 cm long, 25 cm across) built in the mid 1980's by the Mobile Robotics Group at MIT [7]. Genghis is the predecessor to Hannibal; it has six legs, each with 2 DOF (lift and shoulder). Genghis' locomotion network implemented a reflexive controller which produced the step pattern of each leg. In contrast, gait coordination was produced using a CPG-like mechanism. Genghis used only one gait, but more gaits could have been added by modifying the gait sequencer. Brooks programmed Genghis to walk over small obstacles, back away from objects touching its whiskers, and follow people.

Beer and colleagues implemented a neural network control architecture on their small hexapod robot with 2 DOF legs [2,9,29]. The neural controller was developed by Beer and was inspired by Pearson's *flexor* *burst-generator* model of cockroach locomotion. This model incorporates input from a central control signal and sensory information from the legs. At the center of each leg controller is a pacemaker neuron whose output rhythmically oscillates. The pacemaker, and a reflexive mechanism which uses sensory inputs, coordinate the swing phase and support phase of its leg. Inserting mutually inhibitory connections between the pacemaker neurons of adjacent legs generates statically stable gaits, and phase-locking the pattern generators on each side of the body enforces metachronal waves. Using this network, the Case Western Hexapod is capable of producing a continuous range of wave gaits by varying the tonic level of activity of the command neuron. The system could still produce a subset of gaits if either the central control signal or all sensory information was removed.

4.3. Patterned controllers

In simulation, Ref. [10] examines the role of CPGs in producing insect like gaits independent of peripheral sensory input. We consider this a patterned approach since the legs execute programmed motions in response to CPG signals.

Ref. [10] uses coupled non-linear oscillator models to examine inter-limb coordination during locomotion. One goal of this work is to show that general models of simple neuronal oscillator circuits could produce common walking patterns of six legged animals in the absence of sensory feedback. They put forth a network of six symmetrically coupled non-linear oscillators as a possible model for hexapod locomotion CPGs. In this model, the stepping movements of each limb were controlled by a single oscillator. Inter-limb coordination is produced by the coupling and dynamic interactions of the oscillators. Because the CPG oscillator models did not include afferent feedback from leg sensory organs, the resulting timing signals, phase-locked oscillation patterns, and gait transitions are only under central influences. However, the work does not exclude the role of sensory feedback in locomotion. The authors put forth that it is unclear how sensory feedback interacts with the central rhythm generator to adapt the animal's gait to leg perturbations. They suggest that the CPG network continuously produces gait coordination signals, and the afferent signals only modify the effects of the centrally generated coordinating signals at a lower motor level.

5. Controller implementations

This section covers the implementation of three controllers. To mimic insect locomotion, locomotion control is distributed among the six legs. Each local leg controller is responsible for generating the cyclic motion of its leg, and the local controllers influence each other to synchronize and coordinate leg behavior. The controllers range from being strongly reflexive to strongly patterned. None of our implementations are true to detail of any insect locomotion model since modifications were made as needed to run on our robot. However, these implementations capture the reflexive versus patterned nature of the mechanisms.

5.1. The reflexive controller

The reflexive controller exploits stimulus-response reactions to produce locomotive behavior. Our reflexive controller is adapted from Cruse's model of insect locomotion. The motion of each leg and the coordination between legs strongly depends on proprioceptive information. This information is used by three mechanisms (*mechanism*$_1$, *mechanism*$_2$ and *mechanism*$_3$ as presented in Section 3.3) to adjust the excitatory and inhibitory influences that regulate the behavior of leg effectors. Hence, locomotory behavior is achieved from continual computing of sensory inputs and limited motor expressions.

5.1.1. Reflexive individual leg control

In this implementation, each leg has a network of three agents as shown in Fig. 10: a stroke-transition agent, a return stroke agent, and a power stroke agent.

• The step pattern generator (SPG) agent is responsible for switching between the return stroke and the power stroke. The transition from the return stroke to the power stroke occurs when the leg position is greater than or equal to the AEP. In this implementation, the AEP is fixed. The transition from the power stroke to the return stroke occurs when the leg position is less than or equal to the PEP. It computes its PEP from *influence*$_1$,*influence*$_2$ and *influence*$_3$ coming from the peripheral legs as described in Section 5.1.2.

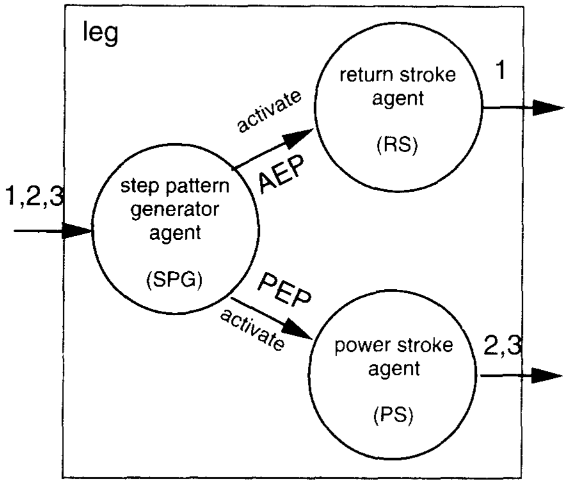


Fig. 10. The cyclic stepping pattern of each leg was implemented using this circuit. The step pattern generator (SPG) agent produces the step cycle by exciting the return stroke agent and the power stroke agent in turn. The arrows with numbers represent the direction the coordination mechanisms were sent between the legs. The numbers represent Cruse's coordination influences as labeled in Section 3.3.

- The return stroke agent is responsible for lifting and swinging the leg to the starting position of the power stroke. It is activated by the SPG agent. While this agent is active, it exerts *mechanism*$_1$. It receives the AEP from the SPG agent.
- The power stroke agent is responsible for supporting and propelling the body by steadily moving the leg to the PEP. Once activated, this agent exerts *mechanism*$_2$ and *mechanism*$_3$. It is activated by the SPG agent. It receives the updated PEP from the SPG agent.

5.1.2. *Reflexive gait coordination*

We implemented the three mechanisms described in Section 3.3 to produce coordinated gaits. Fig. 11 shows the gait coordination circuit. The form of the inhibitory/excitatory influences are shown in Fig. 7. The return stroke agent implements *mechanism*$_1$ by sending a negative constant in the rostral direction during its activation until 60 ms after its deactivation. The power stroke agent implements *mechanism*$_2$ by sending a positive constant in the rostral direction starting 60 ms after its activation and ending 60 ms later. The power stroke agent implements *mechanism*$_3$ by sending a positive monotonically increasing ramp function in the caudal direction during its activation. The new PEP of a leg is computed from the influences it receives by the formula

$$\mathrm{PEP} = \mathrm{PEP}_{\mathrm{default}} + \sum \mathit{Influence}_1 + \sum \mathit{Influence}_2 + \sum \mathit{Influence}_3\,, \qquad (1)$$

where the standard stride length is given by

$$\mathit{Stride}_{\mathrm{standard}} = \mathrm{AEP} - \mathrm{PEP}_{\mathrm{default}} \qquad (2)$$

5.2. *The hybrid controller*

The hybrid controller possesses several biologically motivated mechanisms, but it is not adapted from any particular model of insect locomotion. In our implementation, the local leg control is pattern-like, but the gait coordination is reflexive.

The hybrid controller depends less on proprioceptive information and more on pre-programmed leg motions to produce locomotory behavior. Proprioceptive information is used to generate timing signals to coordinate the pre-programmed leg motions. The timing signals take the form of excitatory and inhibitory messages which are biologically motivated. The mechanisms used to generate these messages are very similar to those used by [19].

5.2.1. *Hybrid individual leg control*

The hybrid controller uses pattern-based mechanisms for individual leg control. Local leg control consists of three agents: the step pattern generator agent, the return stroke agent, and the power stroke agent. The circuit is responsible for generating the cyclic motion of each leg.

- The step pattern generator agent is responsible for switching between the power stroke and the return stroke. It does not transition from the power stroke to the return stroke when the leg position is less than or equal to the PEP. Unlike the SPG agent of the reflexive controller, it does not compute the new PEP from incoming influences. Instead, it makes the transition once it has received an excitatory message and has stopped receiving inhibitory messages from peripheral legs. It transitions from the recover stroke to the power stroke when the leg position is greater than or equal to the AEP.

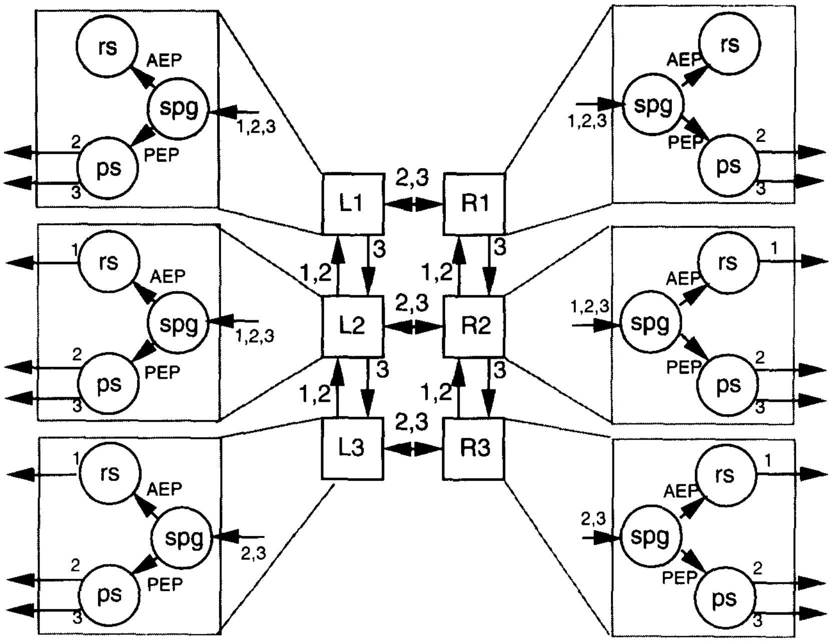


Fig. 11. This circuit implements reflexive gait coordination on Hannibal. A step cycle circuit was implemented on each leg to produce the stepping pattern for that leg. The coordination mechanisms and their configuration (represented as numbers with arrows) are the same as those presented in Section 3.3. Each leg is represented by a black box. The gray box affiliated with each leg illustrates which agents (return stroke or power stroke) send which influences. The influences sent by a leg follow the format shown in Fig. 7.

• The return stroke agent is responsible for moving the leg to the starting position of the power stroke. It is activated by the step pattern generator agent. While this agent was active, it executes a pre-programmed motion that lifts and swings the leg to the starting position of the power stroke. While active it also exerts *mechanism*$_1$ as described below. The AEP is fixed.

• The power stroke agent is responsible for supporting and propelling the body by steadily moving the leg to the PEP. It is activated by the step pattern generator agent. While this agent is active it exerts *mechanism*$_2$ and *mechanism*$_3$ as described below. The PEP is fixed.

5.2.2. *Hybrid gait coordination*

The hybrid controller uses reflex-based mechanisms for gait coordination. Three mechanisms, as shown in Fig. 13, are used to coordinate the robot's gait. Fig. 12 shows how these mechanisms are routed between the legs.

• *hybrid – mechanism*$_1$: During the recover stroke, send an inhibitory message to all adjacent legs. This enforces stability by not allowing adjacent legs to lift at the same time.

• *hybrid – mechanism*$_2$: During the start of the stroke phase, send an excitatory message in the rostral direction. This enforces back to front metachronal waves along each side of the body.

• *hybrid – mechanism*$_3$: During the stroke phase send an excitatory message in the contra-lateral direction of the same segment. This enforces 180 degree phasing between adjacent contra-lateral legs.

5.3. *The patterned controller*

The patterned controller is adapted from Pearson's model of insect locomotion as presented in Ref. [31]. It produces locomotory behavior by coordinating pre-programmed leg motions. Oscillators provide timing information which is used to generate the cyclic motion of each leg and enforce coordination between legs - proprioceptive information is used more to compensate for leg perturbations than to produce basic walking behavior. The inclusion of CPGs makes for an interesting controller, and other researchers are exploring its uses in robot locomotion [7,20,2].

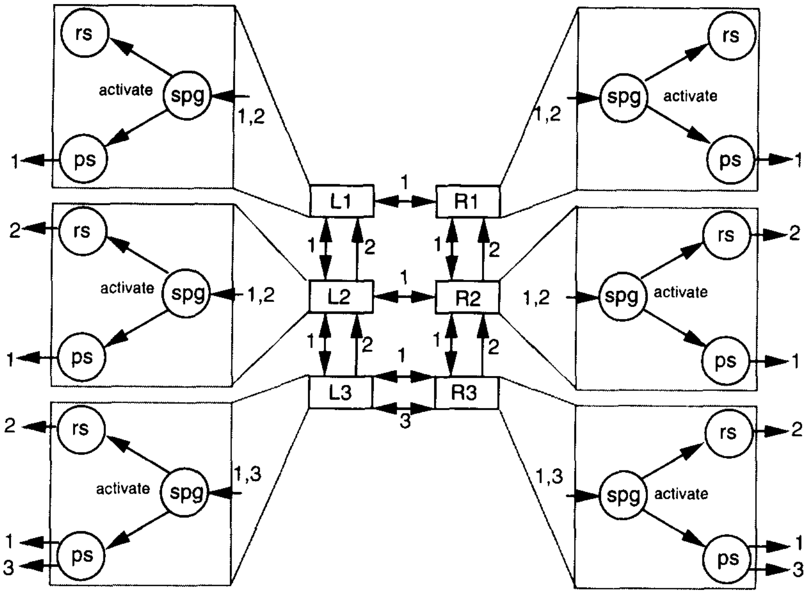


Fig. 12. This circuit implements hybrid gait coordination on Hannibal. A step cycle circuit was implemented on each leg to produce the stepping pattern for that leg. The coordination mechanisms and their configuration (represented as numbers with arrows) are similar to those used in Ref. [19]. Each leg is represented by a black box. The gray box affiliated with each leg illustrates which agents (return stroke or power stroke) send which influences. The influences sent by a leg follow the format shown in Fig. 13.

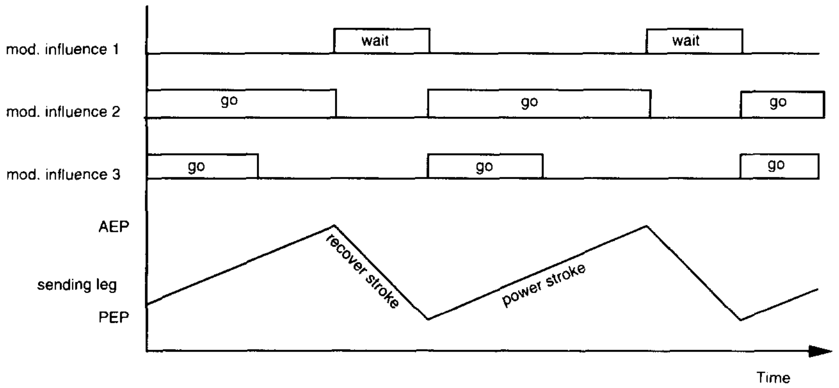


Fig. 13. A diagram showing the relationship between leg position of the sending leg and the coordinating influences it sends. An excitatory influence corresponds to "go", and an inhibitory influence corresponds to "wait". This figure illustrates the time and duration of the coordinating influences as a function of leg position.

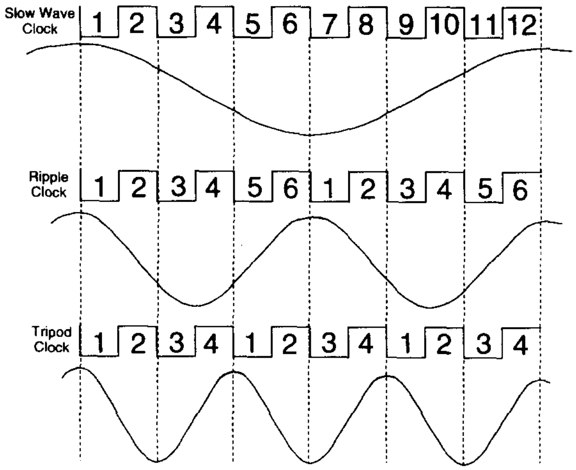


Fig. 14. Implementation of leg oscillators on Hannibal. Each oscillator is modeled as a clock which cycles through its values at regular time intervals. The peak of the oscillator phase corresponds to `osc-clock = 1`. The period of the clock corresponds to the period of the oscillator.

5.3.1. *Patterned individual leg control*

In Pearson's model for individual leg control, a pacemaker on each leg is responsible for generating the cyclic motion of the leg (see Fig. 8). It does this by coordinating the return stroke and power stroke of the leg. Similarly, each of Hannibal's legs has an *oscillator agent* which generates the cyclic motion of the leg. The oscillator agent is modeled as a clock that cycles through its values at regular time steps (see Fig. 14). A *stage* spans one oscillator clock period. We say the oscillator period (the time between adjacent oscillator peaks) spans M clock stages. Each oscillator is synchronized to a common clock. In the current implementation, the oscillator clock value changes every 0.6 seconds.

To produce the return stroke, the oscillator agent excites a *lift agent* (which lifts the leg), a *swing agent* (which swings the leg to the AEP), and a *step agent* (which lowers the leg until it support s the body) as shown in Fig. 15. The lift, swing, and step agents serve a similar function as the flexor neurons of Pearson's model. The lift portion of the recovery phase is performed during the first stage of the oscillator (clock value = 1). The step portion of the recovery phase is performed during the second stage of the oscillator (clock value = 2). During these two stages, the leg swings to the AEP.

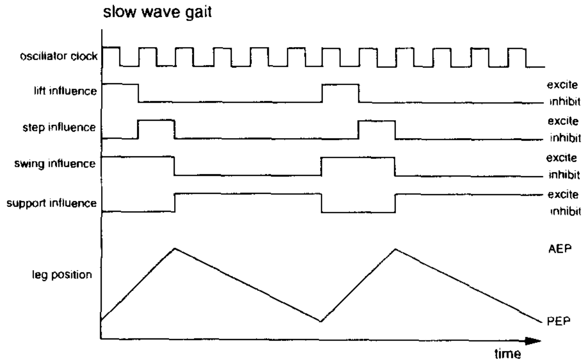


Fig. 15. The influences used in the patterned gait coordination mechanism. The leg lifts and swings for the first clock stage, steps and swings for the second clock stage, and supports and propels the body for the remainder of the oscillator period.

To produce the power stroke, the oscillator agent excites the *support agent* as shown in Fig. 15. The support agent serves a similar function as the extensor neurons of Pearson's model. For clock stages 3 through M, the support agent moves the leg an incremental amount towards the PEP each clock stage, where the increment is equal to S/N where S = abs(PEP − AEP) and N = number of support stages. This increment is determined so that the leg reaches the PEP during the last of the series of support stages. Ergo all supporting legs propel the body in synchrony, and the duration of the power stroke is equal for all the legs. The transition from the support phase to the swing phase occurs on clock value = 1, immediately after the leg reaches the PEP.

5.3.2. *Patterned gait coordination*

Gait coordination is established by properly synchronizing the leg oscillators. As described in Section 3.4, contra-lateral legs of the same segment alternate in phase. Hence, the patterned controller maintains a 180 degree phase difference between contra-lateral oscillators of the same segment. The patterned controller produces back to front metachronal waves by keeping the same temporal lag between adjacent ipsilateral oscillators, where the front oscillator lags the middle oscillator which in turn lags the rear oscillator. The fixed lag between adjacent ipsilateral oscillators keeps the stepping intervals constant between the hind leg and middle leg, and between the middle leg and the front leg. Ipsilateral stability is maintained by constraining the phase difference between adjacent legs such that no leg protracts until the leg behind is placed in a supporting position.

Similar to Pearson's model, Hannibal changes gait as a function of oscillator frequency. An agent called the *speed agent* (analogous to Pearson's command neuron) is responsible for changing the leg oscillator frequency of all the leg oscillators. In addition, local inter-leg communication agents called *coupling agents* are responsible for establishing the phasing between leg oscillators by sending coupling signal messages between local leg controllers. Changing the value of M changes the oscillator frequency. This excites the coupling agents to change the phase between the metachronal waves along each side of the body (Fig. 9), which causes different wave gaits to emerge.

Currently, Hannibal uses three gaits: a slow wave gait, a ripple gait, and a tripod gait. Wilson [31] reports the slow wave gait is the slowest gait and the tripod gait is the fastest gait observed in insects. More gaits could easily be implemented using this network by varying M and the phasing between leg oscillators.

- The Speed agent: Whenever Hannibal wants to change speed, this agent sends a new M value to all the oscillators. This serves to extend the support phase of the step cycle since the recover phase takes a constant amount of time independent of walking speed. Higher level agents command speed through this agent.
- The Activate-Phasing agent: If a new M value is sent to the leg oscillators from the Speed agent, this agent finds the furthest posterior leg performing the lift-and-swing stage (the most rearward leg with its oscillator in the first stage) and activates its coupling agent.
- Coupling agents: When a different M value is sent to the leg oscillators, the instantaneous oscillator clock values must be re-phased with respect to each other to maintain a proper gait. The active coupling agent sends messages to all legs which re-initialize their current oscillator clock values (with respect to the oscillator clock value of the leg with the active coupling agent). Correspondingly, the leg oscillators simply activate the proper agent (lift, swing, step, or support) in accordance to its new clock value, and continue to run as usual.

6. Tests and evaluation

To evaluate the performance of these controllers, we tested each on Hannibal. We ran Hannibal through a series of four experiments of increasing difficulty: walking with unloaded legs, walking on flat terrain, walking with a disabled leg, and walking with external leg perturbations. As well as visually observing the robot's gait, locomotion information was recorded during run-time. As the robot walked, the time history of its gait was sent over a serial port from the robot's on-board computer to a Macintosh computer. The Macintosh computer stored the run-time data to disk so that it could be plotted later.

Each controller was evaluated with respect to gait stability, gait variety as a function of retraction speed, and efficient leg coordination. These evaluation criteria are used to compare the robustness, flexibility, and effectiveness of locomotion produced by each controller. Regarding locomotion robustness, it is of paramount importance that the controller maintain a stable gait in a wide variety of circumstances. Hence, we examined each controller's ability to maintain a stable gait during the four types of experiments. Flexible locomotion is also important. One measure of flexibility is the ability to produce a variety of gaits. A robot could adopt different gaits to accommodate speed, energy consumption, and safety considerations. For example, a tripod gait is faster than a slow wave gait, but a slow wave gait is stable even with the loss of a leg. It is not surprising that insects employ a variety of gaits. Finally, for locomotion to be effective, all the legs must cooperate efficiently. Locomotion can be viewed as a team effort involving all the legs. For example, each supporting leg should contribute to advancing the robot – no supporting leg should be "dead weight" for the others to push along. Furthermore, all supporting legs should propel the body in synchrony – supporting legs should not push at different times. The ability to effectively coordinate the legs becomes more important as the terrain becomes more difficult. Failure to do so could result in instability.

6.1. Air walking

In "air" walking, Hannibal's body is supported on a platform while the legs swing freely. In these simplified conditions, we wanted to determine whether the

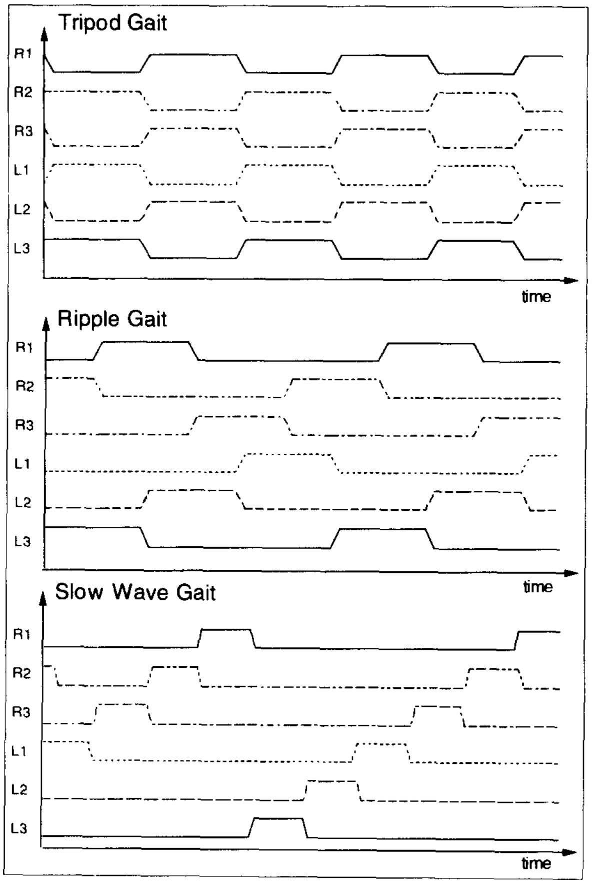


Fig. 16. "Air" walk results using the patterned controller. Run-time data of the three gaits: slow wave gait, ripple gait, and the tripod gait.

controller could produce a variety of stable gaits independent of loading considerations. Each experimental run consisted of at least 20 gait cycles (one gait cycle is equivalent to one complete circuit of each leg performing its step cycle). The series of "air" walking experiments involved both single gait and multiple gait runs. During multiple gait runs, the retraction velocity was varied to evoke different gaits. On average, the retraction velocity was varied 2 - 3 times per multiple gait run.

Each controller produced stable wave gaits for the single gait and multiple gait experiments. The gaits remained stable and efficient throughout the entire run, as the phase of the supporting legs were well synchronized. After some effort in adjusting the reflexive controller parameters, Hannibal performed a wide range gaits as a function of retraction velocity. Hannibal displayed a slow wave gait at the lowest velocity, a tripod gait at the highest velocity, and a variety of intermediate wave gaits for middle range velocities. The transitions between gaits were smooth, and Hannibal maintained a gait until the velocity was changed. We had similar results with the hybrid controller. However it was difficult to produce as many intermediate speed gaits as the reflexive controller. We suspect this is a result of the more stringent transition mechanisms for the hybrid controller. The patterned controller also performed well. We only implemented three gaits for the patterned controller (see Fig. 16), but more could have been added by including different clock periods in the controller.

6.2. *Walking on flat terrain*

Walking on flat terrain introduces loading, friction, and inertia effects on the robot. Stability becomes an important issue since the robot lunges and stumbles when not properly supported. Proper leg coordination is important for efficiency considerations. For example, if the supporting legs push the robot forward at different times, then the legs pushing at a given moment have to work harder to compensate for the legs not pushing at that moment. We included load sensing and ground contact sensing capabilities to each of the controllers. This information was used by the controllers to determine when a given leg was supporting the robot - see Ref. [22] for a complete description.

For each experimental run, the robot walked in a straight line on a smooth, flat floor for approximately 8 feet (about 8 body lengths). We looked at both single gait and multiple gait runs. The retraction velocity was varied about 2-3 times per multiple gait run. We evaluated each controller in terms of maintaining stability and efficient leg coordination as the robot changed gaits. We simply report our findings here, and discuss them in Section 7.

We found the performance of the reflexive controller was affected once the robot was placed on the ground. While running the reflexive controller, Hannibal's gait displayed periods of consistency but would occasionally fall out of phase. On average, the robot's gait fell out of phase about 2 times per run. As a result, the robot had problems maintaining a single gait given a fixed velocity, and it suffered moments of instability

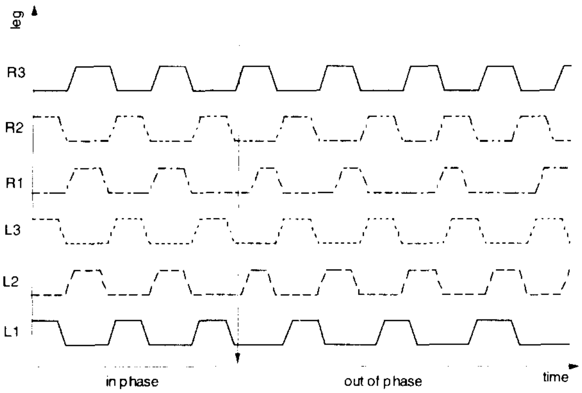


Fig. 17. Flat terrain results using the reflexive controller. We found that inconsistent retraction velocities of the legs cause the robot's gait to shift in and out of phase once the legs are loaded. The robot starts out using the tripod gait, but eventually the legs shift out of phase.

until a new gait fell into phase (as shown in Fig. 17). Performance was improved by adjusting mechanism parameters, but we never achieved the same success as others who have tried a similar approach [4,17], and [30].

The results of the hybrid controller are mixed. On the positive side, Hannibal consistently made gait transitions between stable metachronal gaits ranging from a slow wave gait to a tripod gait (see Fig. 18). The gaits remained stable throughout each run. On the negative side, mid-range gaits were difficult to reproduce, and leg coordination during the support phase was compromised. On occasion some legs would reach the PEP too soon and wait there until told to start the return stroke. For the time period between reaching the PEP and starting the return stroke, the leg does not propel the body. Consequently, the leg effectively becomes "dead-weight" that the other supporting legs have to drag along.

The patterned controller performed the best of the three. The robot continued to smoothly transition between its three gaits, and it maintained its gait for a fixed speed command (see Fig. 19). Each gait remained stable, and the inefficient leg coordination problems of the hybrid controller were resolved by the pacemakers. As described in Section 5.3, the pacemakers synchronize the step cycle stages of the legs. The step cycle stages include lift/swing, step/swing, and a sequence of support stages. During the lift and step stages, the leg moves to the AEP. During each support stage the leg moves an incremental distance towards the PEP. This increment is determined so that the leg reaches the PEP during the last of the series of support stages. Ergo, all supporting legs propel the body in synchrony, and the duration of the power stroke is equal for all the legs.

6.3. *Walking with lesioned leg(s)*

One appealing characteristic of insect locomotion is the ability to maintain a stable gait despite the loss of a leg. Experimental findings regarding the effect of lesions on cockroach gaits is presented in Ref. [31]. He reports when the two middle legs of a cockroach are removed, the animal resorts to a slower gait where a sufficient number of legs are supporting the body at any given time. If the middle two legs of a cockroach are removed, the slow wave gait is still stable but the tripod gait and ripple gait are unstable. Consequently, the cockroach adopts a slow wave gait[3].

For the lesion tests, a single leg was disabled while the robot walked in a straight line on a smooth, flat floor. To enable Hannibal to endure the loss of a leg, we added lesion compensation mechanisms to the hybrid and patterned controllers (the modification to the hybrid controller could easily be made to the reflexive controller since the gait coordination mechanisms are very similar). We examined the cases of disabling a front leg, a middle leg, or a rear leg. We tested the hybrid and patterned controller, but not the reflexive controller, since both reflexive and hybrid tests would be redundant. We evaluated the controllers in terms of their ability to switch to a stable gait once a leg was disabled.

In the hybrid controller, lesion compensation is implemented by treating the step pattern generator of the lesioned leg as a switchboard for the messages of the peripheral legs. An example of this is shown in Fig. 20 where the left middle leg (L2), is the lesioned leg. With L2 gone, the left front leg (L1) and left rear leg (L3) are ipsilaterally adjacent legs. For extra support, L3 is considered contra-laterally adjacent to the right middle leg (R2) and right rear leg (R3). Inhibitory messages of L1 and L3 are routed to each other through L2. Inhibitory messages of L3 and R2

[3] The cockroach adopts one of two possible gaits when the two middle legs are removed. We mention only the slow wave gait.

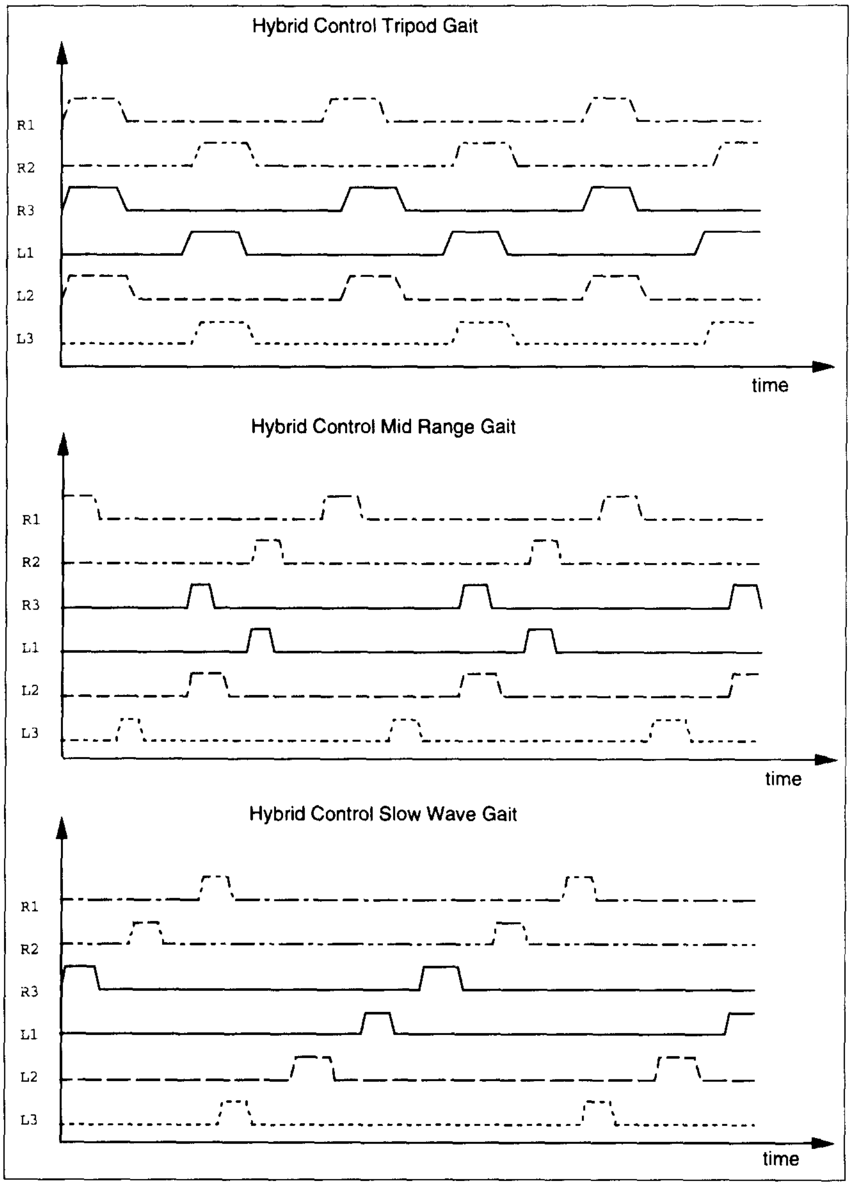


Fig. 18. Flat terrain results using the hybrid controller. Various gaits are produced: the uppermost plot shows a tripod gait, the middle plot shows a mid-range gait, and the bottom plot shows a slow wave gait.

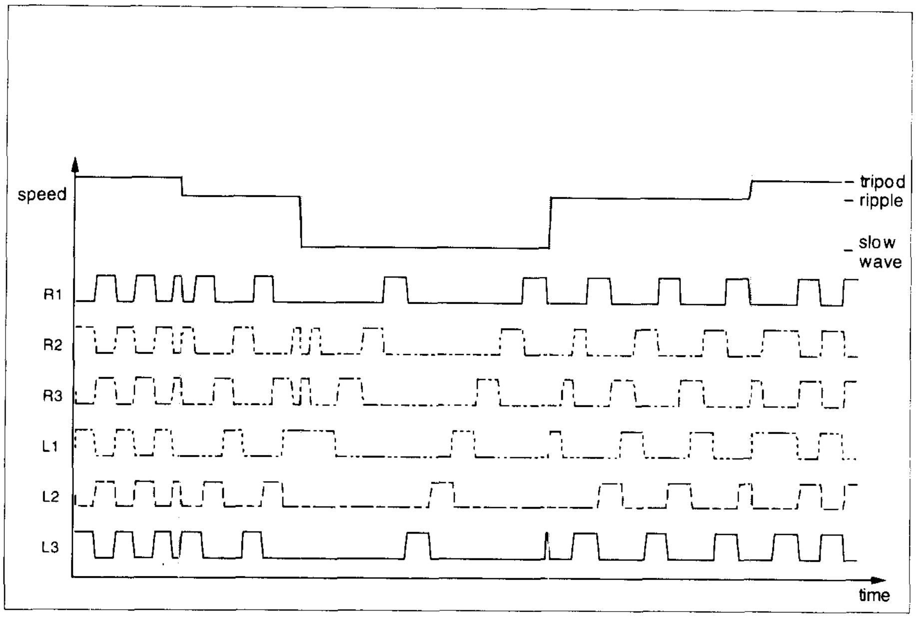


Fig. 19. Flat terrain results using the patterned controller. Run-time data of the robot transitioning between gaits as a function of oscillator period.

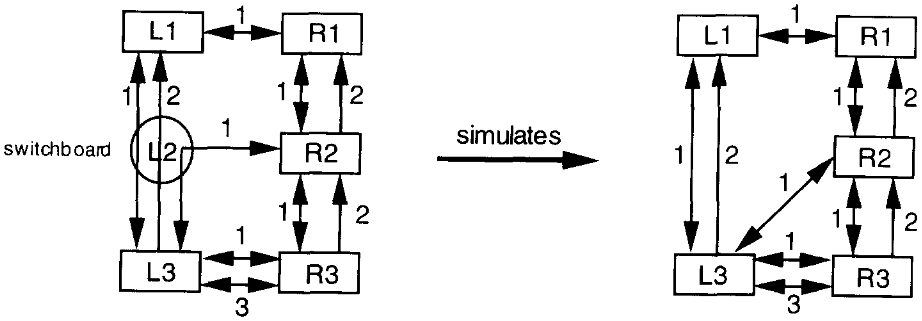


Fig. 20. By re-routing the leg coordination influences through the lesioned leg, gait coordination was maintained despite the loss of the leg. This modification was incorporated into the hybrid controller. The same modification could be made to the reflexive controller.

are also routed to each other through L2. The excitatory message of L3 is routed through L2 to L1. Effectively, the influences of the lesioned leg are removed from the network and replaced by the influences of its adjacent legs. Fig. 21 shows how re-routing the coordination influences through the lesioned leg affects the gait. As a result, the robot immediately switched to a slow wave gait when any single leg was disabled. Hence, the gait remained stable during the run despite the loss of a leg.

Pearson's model for insect locomotion (the version presented in this paper) does not account for lesion compensation. Consequently, we added a lesion mechanism to the patterned controller. In our implementation, an additional process is added to each local leg controller. This process becomes active if its leg becomes in-operable – [21] discusses how the controller determines when a leg is useless. When active, it disables its damaged leg by putting all of its motors into brake mode and sends a message to the *speed agent* to evoke the slowest wave gait. Although this mechanism is not biologically motivated, it proved effective

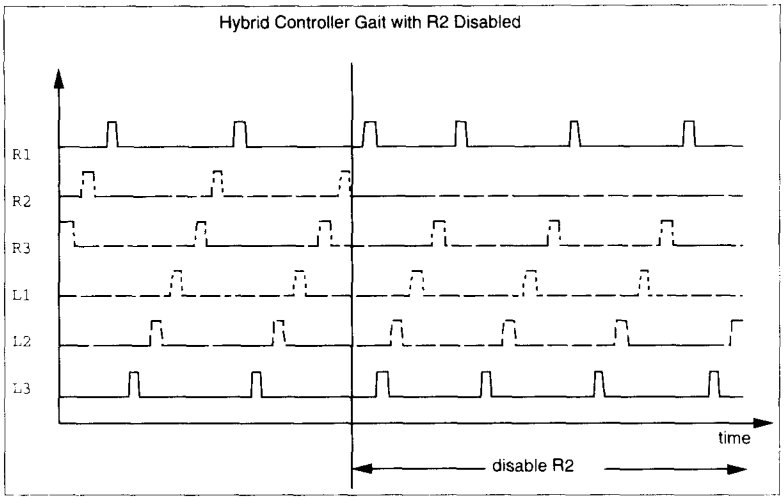


Fig. 21. Lesion test results of the hybrid controller. Run-time data of the robot's gait before and after the right middle leg (R2) is lesioned. Notice that after leg R2 was removed, leg R1 began its recovery phase immediately after leg R3 finished its recovery phase.

and easy to incorporate (recall that our bias is to develop a flexible and robust locomotion controller for our robot – not to limit ourselves to biologically plausible mechanisms). The robot's gait switched to the slow wave gait immediately after the "lesion", and the gait remained stable for the remainder of the run.

6.4. Walking with leg perturbations

Insect locomotion is extremely robust to a variety of gait disturbances. A long time goal of legged locomotion is to build walking machines capable of traversing rough terrain like their biological counterparts. [28] observed that locusts have a elevator reflex when their legs strike objects during the swing phase. As described by Pearson, the elevator reflex is "a rapid elevation and extension of the leg to lift it above the object (provided the object is not too high), followed by depression, which usually results in the object being used as a site of support". We incorporated collision sensing abilities and an elevator reflex behavior into each controller so the robot could step over small obstacles (a complete description of these processes is described in Ref. [22]).

We tested how each of our locomotion controllers responded to a variety of leg perturbations such as delaying a leg during the swing phase, delaying a leg during the support phase, and walking over irregular terrain. In our tests, delaying a leg involved manually grabbing and holding it still for a fraction of the time required for the leg to finish its current step cycle phase. Irregular terrain consisted of a small obstacle (such as a book or rock) obstructing the robot's path. A small obstacle is less than 1/3 of the robot's overall height. We present our findings here and discuss them in detail in Section 7.2.

We tested the reflexive controller while Hannibal "Air" walked so that we could isolate stability problems due to loading from instability due to externally applied leg perturbations. This controller enabled Hannibal to sustain only minor gait disturbances. The controller maintained a stable gait despite short delays applied during the swing or stance phases. During the stance phase, a delay lasted for about 1/8 of total retraction time, and for the recover phase, a delay lasted for about 1/4 of the total swing time. In general, locomotion stability was more sensitive to stance phase delays, and the front legs were more resilient to delays than the other legs. If a leg was not allowed to reach its computed PEP, it would miss its return stroke. This had a more dramatic effect on gait coordination for more posterior legs. If delays caused gait instability, gait coordination was typically regained within 4 – 8 step cycles. To simulate collisions with a small object, we manually obstructed the leg in mid swing to evoke the elevator reflex. The gait always became unstable during this time.

The hybrid controller was tested while Hannibal walked on the ground. This controller also maintained stability despite short delays during the swing or support stages. The gait was more robust to delays during the stroke phase than the reflexive controller. However, the mechanism seemed vulnerable to a "halting chain reaction" when subjected to long delays. If any leg was held fixed such that it missed its step cycle, that leg would fail to initiate the step cycle of its anterior leg. This set up a back to front chain reaction where each successive leg would fail to perform its step cycle. Once the robot came to a complete stop, it would not start again until the controller was reset. When walking on irregular terrain, the gait became unstable when the robot tried to step over a small obstacle.

In contrast to the reflexive and hybrid controllers, the patterned controller maintained stability despite a wide assortment of leg perturbations. Short delays applied during the swing and support stages did not affect locomotion stability since the transitions between the support and swing stages were tightly regulated by the oscillators, and the phasing between oscillators maintained proper gait coordination. To handle long delays, such as those encountered when Hannibal steps over obstacles, we added a mechanism which modulates leg oscillator state transitions. The mechanism is quite simple: while a leg is performing the swing phase, the oscillators of the supporting legs are temporarily inhibited. This effectively keeps the supporting legs in their current oscillator state. As described in Section 5.3, the state of each oscillator determines the retraction position of its leg. These positions are computed so that the supporting legs are in a stable configuration at all times. Thus stability is maintained while the robot waits for its leg to clear the obstacle. The inhibitory influence is removed when the swinging leg regains ground contact. This is not a biologically motivated mechanism, but it is effective and easy to incorporate into our controller.

In general, it is important for a controller to readily support incremental design - such as when adding more sophisticated rough terrain abilities to the basic locomotion repertoire. In addition to exhibiting gait stability, flexibility, and efficiency, the patterned controller is easy to add new behaviors to. We eventually added a wide assortment of terrain characterization processes and rough terrain behaviors to the patterned controller. These processes allow the robot to avoid cliffs, walk over gaps, handle sloping terrain, and traverse obstacles of different sizes. We refer the interested reader to [22] which offers a detailed presentation Hannibal's rough terrain implementations and performance.

Table 1
Controller comparison summary. If the gait stability was poor for a given test, then gait variety and gait efficiency could not be measured. Gait variety is not measured for the lesion test since only one gait is stable with the loss of a leg. Gait efficiency was not measured for the lesion test since gait efficiency is automatically diminished with the loss of a leg.

Controller	Leg perturbation	Gait stability	Gait variety	Gait efficiency
Reflexive				
	unloaded	good	good	good
	loaded	fair	good	fair
	external	poor	N/A	N/A
Hybrid				
	unloaded	good	fair	good
	loaded	good	fair	fair
	lesion	good	N/A	N/A
	external	poor	N/A	N/A
Patterned				
	unloaded	good	fair	good
	loaded	good	fair	good
	lesion	good	N/A	N/A
	external	good	fair	good

7. Discussion

We found the robot performed quite differently depending upon which controller was used (see Table 1). In this section, we discuss design issues that most affected the performance of our controllers. These design issues include consistent retraction velocity between legs, leg design, sensing capabilities, and computational requirements.

7.1. *Sensitivity to retraction velocity*

Our results suggest that reactive gait coordination schemes are more sensitive to inconsistent retraction velocities between supporting legs than patterned gait coordination schemes. While using reactive gait coordination mechanisms, we found that inconsistent retraction velocities among supporting legs led to unstable and inefficient locomotion. n [9] mentions similar experiences using reactive gait coordination mechanisms. When their robot relied only on sensory feedback, it could generate statically stable gaits provided the legs were moved at a constant velocity. In addition, they found that the robot's gait became unstable if its legs moved through different angle ranges. Inconsistent retraction velocities would cause the supporting legs to travel through different sized angles thereby inducing gait instability. [17] is currently extending his simulation to take into account legs moving at different velocities during the power stroke.

Our reflexive controller had difficulty maintaining gait stability once Hannibal was put on the ground. In strongly reflexive control, leg coordination emerges from adjusting the PEP of the legs. The transition to the swing phase occurs when each leg reaches its PEP. It is important for all supporting legs to retract at the same velocity for the swing transitions to occur at the proper time. Once Hannibal was placed on the ground, its legs did not retract with the same velocity. This is because Hannibal's velocity control cannot compensate for loading and friction effects on the legs. As a result, the legs would occasionally transition to the swing phase at the wrong time. This caused Hannibal to change gaits randomly and occasionally become unstable (as shown in Fig. 17).

In contrast, the hybrid controller forces the start of the recover stroke as soon as the adjacent legs are supporting the body and the adjacent posterior leg finishes its recover stroke. In the special case of the hind legs, the return stroke starts when the leg reaches the PEP, all adjacent legs are in the supporting position, and the adjacent contra-lateral leg has completed its return stroke. Consequently, the hybrid approach is less sensitive to retraction velocity and is better suited to Hannibal. The hybrid control approach produces a range of stable gaits as shown in Fig. 18. But poor retraction velocity control presents a different problem. Because Hannibal cannot consistently control the duration of the power stroke, some legs reach the PEP too soon and wait until told to start the return stroke. For the time period between reaching the PEP and starting the return stroke, the leg does not propel the body. Consequently, gait coordination becomes inefficient.

Given Hannibal's problem with maintaining consistent retraction velocities, the patterned controller worked the best. As described in Section 5.3, the patterned controller avoided velocity control induced problems by using oscillators. The oscillators maintain gait stability by carefully regulating the step cycle transitions between legs. To prevent inefficient leg coordination, the pacemakers synchronize the support phase motions. During each support stage, the leg moves an incremental distance towards the PEP. This increment is determined so that the leg reaches the PEP during the last of the series of support stages. Ergo all supporting legs propel the body in synchrony, and the duration of the power stroke is equal for all the legs.

7.2. *Sensitivity to leg design*

7.2.1. *Fast elevator reflex*

Neither the reflexive controller nor the hybrid controller maintained stability while the robot walked over irregular terrain. This is largely a result of Hannibal's painfully slow elevator reflex, not a direct consequence of reflexive gait coordination mechanisms.

Reflexive gait coordination schemes can be effective at walking on irregular terrain provided the robot has a fast elevator reflex. The simulation presented in Ref. [17] handles flat terrain with small objects, and a similar capability was demonstrated on the robot designed by [30]. This robot has a very rapid elevator reflex, so it can step over a small object without dramatically lengthening the time to perform the swing motion. In contrast, Hannibal's legs have extremely high gearing on the lead screw actuating the extension degree of freedom. As a result, it takes Hannibal an extremely long time to step over an obstacle. If the supporting legs continue to retract while the leg steps over an object (as they do for reflexive gait coordination), the robot eventually becomes unstable. If the supporting legs reach their PEP threshold, they begin their swing phase while the leg is still clearing the obstacle. Otherwise the supporting legs all move to an extreme posterior position (which is an unstable configuration) while the leg is clearing the obstacle.

Given Hannibal's slow elevator reflex, the patterned controller worked the best for traversing irregular terrain. As discussed in Section 6.4, the controller modulates the leg oscillators so that the robot remains stable while its leg clears an obstacle.

7.2.2. *Curved leg movements*

Hannibal's leg design causes the legs to move in fairly tight arcs. The upper leg link is short which gives the swinging motion a tight radius of curvature. It was unacceptable to use the extension degree of freedom to compensate for this curved motion because it causes the robot to walk at unacceptably slow speeds.

Curved walking introduces unwanted forces to the robot's legs and body. All supporting legs of the robot are mechanically coupled through the robot and through the terrain. If the legs move along the body in curved trajectories, torquing effects are applied to legs and the body. Torquing effects exacerbate Hannibal's retraction velocity problems since these forces cause the legs to propel the body at different rates and at different times. As argued earlier, this can lead to instability when using reflexive gait coordination mechanisms. The Case Western Hexapod has 2 DOF legs, so it can only perform linear foot trajectories along the body. This may have helped them successfully implement their reflexive gait coordination schemes. Cruse [17] intends to extend his simulation to consider curve walking.

7.3. Sensing requirements

Reflexive mechanisms rely the most on afferent feedback from the legs. As discussed previously, reflexive schemes require sensory input for stable locomotion since leg transitions depend on leg position (i.e. calculating a leg's PEP threshold, and knowing when a leg reaches this threshold). This leads to other requirements such as having the sensory information from the legs reach the computer running the controller with a consistent time delay for each leg, and with a common time delay for all legs, otherwise gait instability occurs [29]. For the strongly reflexive controller, the propulsive motion of each leg cannot be pre-programmed since stroke length changes during run-time. Consequently, both step cycle generation and gait coordination requires continual sensory information. The hybrid controller depends somewhat less on afferent inputs since the AEP and PEP thresholds are fixed. The swing motion is the same no matter what gait is used, so it is effectively pre-programmed. The retraction motion is also pre-programmed for the same reason. However, the transitions between swing and stroke phases needs afferent feedback from the legs to produce stable gait coordination. The patterned controller requires the least sensing of the three. In patterned control, commanded leg motion maps linearly onto oscillator phase, so sensory information is not used for making swing/stroke phase transitions. Some sensory information is used during the power stroke to maintain efficient propulsion. However, once the stride length is known, this information is fixed.

Generally, all types of controllers benefit from afferent feedback. The importance of proprioceptive information is well documented for insect locomotion, although there is some argument regarding how this information is used. For example, patterned schemes tend to use sensory information to modify CPG generated leg motions [10]. Our patterned controller uses this approach for irregular terrain locomotion. Reflexive schemes use afferent feedback to produce leg motion at all levels of control. This was demonstrated by Cruse's controller when walking over irregular terrain [17]. Ref. [9] discusses how reflexive and patterned techniques can be combined to enhance the robustness of gait coordination. Nevertheless, ample afferent feedback is necessary to locomote over rough terrain no matter which control scheme is used. Furthermore, more difficult and varied terrain requires more sensing capabilities so that the controller can characterize it and evoke the correct behavior to respond to it.

7.4. Computational issues

The appeal of reflexive schemes, like Cruse's model of insect locomotion, is their elegance. The authors of Refs. [19,17,2,9] have demonstrated in simulation or on robots that these sorts of controllers provide a lot of functionality for a handful of clever mechanisms. Several interesting behaviors emerge from these controllers including the ability to produce a family of stable insect-like gaits, robustness to lesions, and resiliency to leg perturbations and irregular terrain. We demonstrated these capabilities on Hannibal as much as its leg design would allow.

However, reflexive controllers have drawbacks as well. First, the mechanism parameters require a fair amount of adjustment to achieve acceptable performance. Cruse [17] adjusted his simulation parameters based on results from studying stick insects. However, other researchers have arrived at the proper weighting of influences through optimization techniques [3], or by experimentation [29]. Nonetheless, our reflexive controller never gave us acceptable results largely due to robot design issues.

Another drawback of reflexive controllers is their difficulty to add new behaviors. The controller mechanisms interact in a very specialized way to produce gait coordination. Consequently it is difficult to incorporate dramatically different behaviors without disrupting controller dynamics. For example, locusts use rapid searching movements with an unsupported leg to locate a secure foothold [28]. This behavior is characterized by rapid, repeated cycling of a single leg while the other legs may even come to rest. It is not clear how to modify the current mechanisms to incorporate such an asymmetrical behavior. A typical approach is to simply add some sort of centralized process on top of the locomotion network that can override it [19], but this detracts from the elegance of the reflexive approach.

Finally, there is a long way to go before robots using reflexive control schemes can locomote like insects over naturally occurring terrain. Typical mechanisms do not account for many of the rough terrain locomotion behaviors observed in insects. Pearson found that locusts use several single leg tactics to deal with terrain perturbations such as rhythmic searching movements when a leg does not contact a surface at the end of its swing, an elevator reflex to lift a leg above an object contacted during the swing phase, and local searching movements once a leg has contacted a potential supporting surface. Pearson also found that animals do not adopt a rigid gait when walking on rough terrain. In fact, he observed a wide range of stepping patterns in which the gait pauses and shifts frequently. Progress is being made in this area as evidenced by [17,30], but much work has to be done.

In contrast, the appeal of a patterned approach is the ease of adding new features to the system. On the downside, this makes for a bulkier controller – more code must be added which takes up more computing cycles. But the designer also has more control over the interactions between processes within the controller. This makes the controller easier to design around and build upon, although traditional issues like behavior conflict resolution must be carefully resolved. We have implemented a wide assortment of rough terrain behaviors and terrain sensing processes to our basic patterned-based locomotion controller. Several of these behaviors are inspired after those observed in insects such as local searching movements, an elevator reflex, stepping over holes, walking around large obstacles, and walking away from ledges. These behaviors are described in detail in Ref. [22].

8. Concluding remarks

More and more researchers are interested in insect inspired approaches for hexapod locomotion. Locomotion control involves both the movement of individual legs as well as the coordination between legs. In this work, two approaches were considered: a reflexive approach where leg motion is largely determined by the sensory state of the leg and its adjacent legs, and a patterned approach where leg motion consists of pre-programmed patterns of behavior orchestrated by CPG activity. Typical insect locomotion control schemes implemented on robots can be characterized as being reflexive, patterned, or some of each.

We designed three biologically inspired locomotion controllers (which are fairly representative of the field), and implemented them on the same robot. The flat terrain capabilities of these controllers were motivated by models of insect locomotion. However, modifications were made to the insect models so that our implementations would run on our platform. We added some new mechanisms to the flat terrain controllers so that the robot could accommodate disabled legs or step over obstacles blocking its path. Not all of these modifications are biologically plausible, however it gave us an opportunity to see how easily new capabilities could be added to each controller. These decisions reflect our bias of using insect locomotion models to enhance robot locomotion, as opposed to using robots to study insect locomotion.

By implementing different locomotion controllers on the same robot, we could compare their performance on a common platform while performing a common set of tasks. The chosen tasks are fairly representative of those tasks others use to evaluate their insect-like controllers. We found the robot's performance differed depending upon which controller was used. Through our tests, we uncovered several issues that influenced the controllers' effectiveness.

Two of these issues involve robot design: sensitivity to inconsistent retraction velocity and leg design. These results reveal the intimate interplay between the robot itself and the algorithms implemented on it. Put simply, the robot to be designed influences the software implemented upon it, and the software to be implemented strongly influences robot to be designed. For example, we had poor success with the reflexive gait coordination schemes because of robot design incompatibilities.

As an extreme case in point, we take the CMU Ambler. The Ambler is an autonomous hexapod built at CMU in the early '90s [24-26]. The Ambler is huge, standing approximately 18 feet tall and weighing about 2500 kg. The control concerns of Hannibal are diametrically opposed to the control concerns of the Ambler. Many of these differences result from the drastically different scale of the two physical systems. Because the Ambler stands 18 ft tall, stability is of critical importance to the Ambler. To insure stability, the Ambler plans every step and body movement with great care. It builds terrain maps so the Ambler can carefully select its footholds. Foot placements are chosen so the center of mass of the system always remains in the conservative polygon of support. Foot placement forces are analyzed in detail before and after the step is taken to insure the terrain supports the robot, and the foot will not slip. Leg trajectories are planned so no leg collisions occur during the return stroke, and only one leg performs a return stroke at a time. Body altitude and attitude are carefully monitored so the system does not become unstable. The algorithmic requirements for this robot is practically the antithesis of insect locomotion, and it would be foolish to build such a robot if one wants to study insect-like rough terrain locomotion.

We also encountered some important implementation issues. Two of these issues involve "user friendliness": the difficulty in getting the controller to perform well on a system and the ease of adding new capabilities to a system. In our application, rough terrain locomotion was more important than implementing accurate models of insect locomotion. For example, we could have implemented more biologically plausible methods of phase locked leg oscillators, and there is a lot of work that has been done on this problem [2,10]. We probably would have seen more gaits emerge if we did this, but it would have been much more difficult implementing effective rough terrain behaviors as well.

Given our experience, we see work related to insect motivated controllers for robots moving in two directions. From a scientific standpoint, researchers will continue to refine their insect models of locomotion to resolve the reflexive versus patterned debate, to account for rough terrain behavior, and to incorporate locomotion dynamics. From an engineering research perspective, researchers will continue to use robots to test models of insect locomotion as well as continue work in building better rough terrain walking machines. New hexapod robots are beginning to incorporate insect inspired design concepts. For example, the leg design presented in Ref. [30] is strongly modeled after the walking stick insect. In addition, a small pneumatically actuated robot has been built at MIT where the design of each leg pair was inspired by the structure and function of cockroach legs [5]. In Ref. [23] it was found that cockroaches use each pair of legs differently in locomotion: the front legs are used primarily for probing, the middle legs for support and balance, and the hind legs for propulsion. Binnard's pneumatic actuators provide significantly more force per weight than DC servo motors, so it may be possible to use either this or similar robots to study insect dynamic walking. This has yet to be seen. However, building more insect-like hardware is a step in the right direction for studying insect locomotion control.

Acknowledgements

This work was conducted under the supervision of Professor Rodney Brooks at the MIT Artificial Intelligence Laboratory. Special thanks to Rod Brooks, Mike Binnard for reviewing earlier versions of this work and providing useful comments.

References

[1] C. Angle and R. Brooks, Small Planetary Rovers, *in* Proceedings of IEEE International Workshop on Intelligent Robots and Systems, Ibaraki, Japan (1990), pp. 383-388.

[2] R. Beer and H. Chiel, Simulations of Cockroach Locomotion and Escape, *in* Biological Neural Networks in Invertebrate Neuroethology and Robotics, Academic Press, Inc., Boston (1993) pp. 267-286.

[3] R. Beer and J. Gallagher, Evolving Dynamical Neural Networks for Adaptive Behavior, *Adaptive Behavior* 1 (1992) 91-122.

[4] R. Beer, H. Chiel, R. Quinn and K. Espenschied, Leg Coordination Mechanisms in Stick Insects Applied to Hexapod Robot Locomotion, *in* IEEE International Conference on Robotics and Automation (1992).

[5] M. Binnard, Design of a Small Pneumatic Walking Robot, *Masters Thesis, MIT* (1995).

[6] R. Brooks, A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation* RA-2, 14–23 (1986).

[7] R. Brooks (1989), A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network, *Neural Computation* 1:2(1989) 365-382.

[8] R. Brooks, The Behavior Language; User s Guide, *MIT A. I. Memo 1227* (1990).

[9] H. Chiel, R. Quinn, K. Espenschied, and R. Beer, Robustness of a Distributed Neural Network Controller for Locomotion in a Hexapod Robot, *IEEE Transactions on robotics and automation* 8 (1992) 293-303.

[10] J. Collins and I. Stewart, Hexapodal Gaits and Coupled Non-linear Oscillator Models, *Biological Cybernetics* 68 (1993) 287-298.

[11] H. Cruse, The Control of Body Position in the Stick Insect (*Carausius morosus*), when Walking Over Uneven Terrian, *Biological Cybernetics* 24 (1976) 25-33.

[12] H. Cruse, The Function of the Legs in the Free Walking Stick Insect, *Carausius morosus*, *Journal of Comparative Physiology* 112 (1976) 235-262.

[13] H. Cruse, A New Model Describing the Coordination Pattern of the Legs of a Walking Stick Insect, *Biological Cybernetics* 32 (1979) 107-113.

[14] H. Cruse (1980*a*), A Quantitative Model of Walking Incorporating Central and Peripheral Influences: I. The Control of the Individual Leg, *Biological Cybernetics* 37 (1980) 131-136.

[15] H. Cruse, A Quantitative Model of Walking Incorporating Central and Peripheral Influences: II. The Connections between the Different Legs, *Biological Cybernetics* 37 (1980) 137-144.

[16] H. Cruse, What Mechanisms Coordinate Leg Movement in Walking Arthropods, *Trends in Neurosciences* 13 (1990) 15-21.

[17] H. Cruse, A Neural Net Controller for a Six-Legged Walking System, *in* Proceedings of From Perception to Action, Lausanne, Switzerland, (1994) pp. 55-65.

[18] J. Dean, Coding Proprioceptive Information to Control Movement to a Target: Simulation with a Simple Neural Network, *Biological Cybernetics* 63 (1990) 115-120.

[19] M. Donner, *Real Time Control of Walking*, Birkauser, Boston (1987).

[20] C. Ferrell, Robust Agent Control of an Autonomous Robot with Many Sensors and Actuators, *MIT Artificial Intelligence Lab Technical Report 1443* (1993).

[21] C. Ferrell, Failure Recognition and Fault Tolerance of an Autonomous Robot, *Adaptive Behavior* 2 (1994) 375-398.

[22] C. Ferrell, Global Behavior via Cooperative Local Control, *Autonomous Robots* 2:2, *forthcoming* (1995).

[23] R. Full, Integration of Individual Leg Dynamics with Whole Body Movement in Arthropod Locomotion, *in* Biological Neural Networks in Invertebrate Neuroethology and Robotics, Academic Press, Inc., Boston, pp. 267-286 (1993).

[24] E. Krotkov and R. Simmons, Performance of a Six-Legged Planetary Rover: Power, Positioning, adn Autonomous Walking, *in* Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, pp. 169-174 (1992).

[25] E. Krotkov, R. Simmons and C. Thorpe, Single Leg Walking with Integrated Perception, Planning, and Control, *in* Proceedings of IEEE International Workshop on Intelligent Robots and Systems, Ibaraki, Japan, pp. 97-102 (1990).

[26] P. Nagy, W. Whittaker and S. Desa, A Walking Prescription for Statically-Stable Walkers Based on Walker/Terrain Interaction, *in* Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, pp. 149-156 (1992).

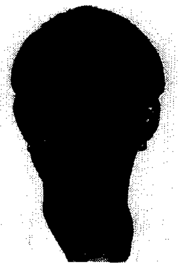
[27] K. Pearson, The Control of Walking, *Scientific American* 235 (1976) 72-86.

[28] K. Pearson and R. Franklin, Characteristics of Leg Movements and Patterns of Coordination in Locusts Walking on Rough Terrain, *International Journal of Robotics Research* 3 (1984) 101-112.

[29] R. Quinn and K. Espenschied, Control of a Hexapod Robot Using a Biologically Inspired Neural Network, *in* Biological Neural Networks in Invertebrate Neuroethology and Robotics, Academic Press, Inc., Boston, pp. 267-286 (1993).

[30] H. Weidemann, F. Pfeiffer, and J. Eltze, A Design Concept for a Legged Robots Derived from the Walking Stick Insect, *in* Proceedings from Intelligent Robots and Systems, Yokohama, Japan (1993) pp. 545-552.

[31] D. Wilson, Insect Walking, *Annual Review of Entomology* 11 (1966).



Cynthia L. Ferrell is pursuing her Ph.D. degree at the Massachusetts Institute of Technology Artificial Intelligence Laboratory in Cambridge, Massachusetts. Her research interests include fault tolerant autonomy, locomotion control, emergent behavior, and adaptation of complex autonomous agents. Ms. Ferrell received her B.S. degree (1989) from the University of California, Santa Barbara in Electrical and Computer Engineering. She received her M.S. degree (1993) from the Massachusetts Institute of Technology in Electrical Engineering and Computer Science.