Proceedings 2005 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
June 27-30, 2005, Espoo, Finland

Legs that can walk: Embodiment-Based Modular Reinforcement Learning applied

David Jacob, Daniel Polani, Chrystopher L. Nehaniv
Adaptive Systems Research Group, University of Hertfordshire
College Lane, Hatfield, Herts AL10 9AB, UK
D.Jacob, D.Polani, C.L.Nehaniv@herts.ac.uk

Abstract— **Experiments to illustrate a novel methodology for reinforcement learning in embodied physical agents are described. A simulated legged robot is decomposed into structure-based modules following the authors' EMBER principles of local sensing, action and learning. The legs are individually trained to 'walk' in isolation, and re-attached to the robot; walking is then sufficiently stable that learning in situ can continue. The experiments demonstrate the benefits of the modular decomposition: state-space factorisation leads to faster learning, in this case to the extent that an otherwise intractable problem becomes learnable.**

I. Introduction

Reinforcement Learning (hereafter RL), in its various forms, has long proven to be useful in robot control of various kinds [13], [16], [19]. However, there are cases where a problem is too complex to be tackled all at once. Given such a problem, some means of decomposing it is required to allow learning to proceed. In robots, for example, there may be several sub-systems, each one of which requires the other sub-systems to act more or less correctly so as to provide a framework within which learning can take place.

Our EMBER (from EMbodiment-Based modulaR) reinforcement learning framework, introduced in [10] developed in [11] and briefly recapitulated in section V below, bases this decomposition on the physical structure of embodied agents. The underlying idea is that many embodiments have a form which naturally suggests a modular decomposition: multiple limbs for example could be treated as separate modules, possessing their own sensors, actuators and learning capability, and able to make fully, or partly, autonomous decisions about actions. In turn, this modularity induces a factorisation in the learning space; this gives a number of advantages in learning speed and the capacity for generalisation.

Our previous work made use of gridworlds for the purposes of proof-of-concept; we now apply the same principles to a more realistic existing scenario to illustrate several facets of our approach.

The problem we have elected to address is legged locomotion. To overcome the difficulty of initial learning, we train legs individually to 'walk' in isolation; once trained, several legs are attached to a body structure, and the whole assembly should then be able to walk well enough to continue to learn to improve its walk. This two-stage learning is made possible because, following EMBER principles, the legs can sense, act and learn locally, the same local sensing scheme being used in both training and walking stages. Once assembled onto the walker, basic walking can then occur with no communication between the legs apart from for synchronisation purposes. Further, although these experiments took place within a simulated environment (albeit one having a degree of physical realism), only sensing which could reasonably be implemented on an actual robot was permitted.

There is unavoidably a considerable degree of 'mechanical' design in such a system and this is briefly introduced next. The sensing and reward schemes together with the learning process are then described. Finally we assess the performance of the complete system and outline future work aimed at providing central overall control of the modules.

II. Mechanical Design of Leg

(The models described were implemented in the ODE virtual physical environment [20] which provides a fair degree of physical realism with the ability completely to integrate controllers of any design.)

Nature has provided us with many models for the design of walking structures. However these tend to be very complicated both in mechanical and behavioural terms. Since the main thrust of this work is in reinforcement learning rather than mechanical engineering, we have devised a much simpler model to illustrate the principle. Nonetheless, the complexity of natural solutions exists for a reason: legged locomotion is intrinsically a difficult problem to solve well and a simplified leg design does not make the underlying problem easier.

A two joint leg was chosen (figure 1). Each joint is equipped with a motor which has controls for speed and torque: the latter sets the maximum torque which will be applied to try to achieve the desired speed. Under load there is therefore interaction between the movements of the two joints which provides a degree of compliance in the system.

Limits were set on the motion of the joints based on a likely operating height which was projected to be 90% of the fully-extended length of the leg. Because it greatly simplifies the overall design to have the knee joint only able to bend backwards (like the human leg) there will be a portion of the stance phase [1] where lift force is transmitted to the body. If the operating height is too low, this lift requires a lot of torque in

[1] the reciprocating action of a leg is conventionally divided into two phases known as 'stance' where the foot is in contact with the ground, and 'swing' where the foot is lifted and moved into position ready for the next step

0-7803-9355-4 / 05 / $20.00 ©2005 IEEE.

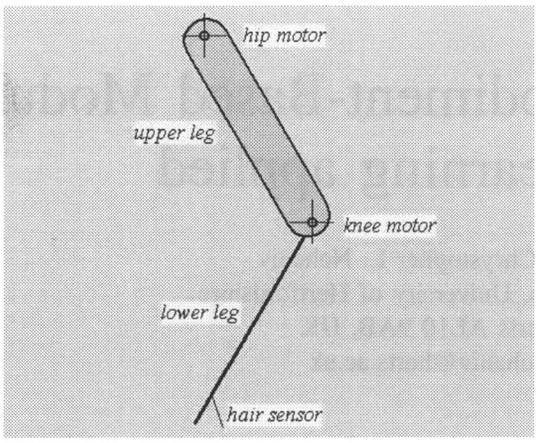

Fig. 1. The two-joint leg. The hair sensor is described in section VI-B

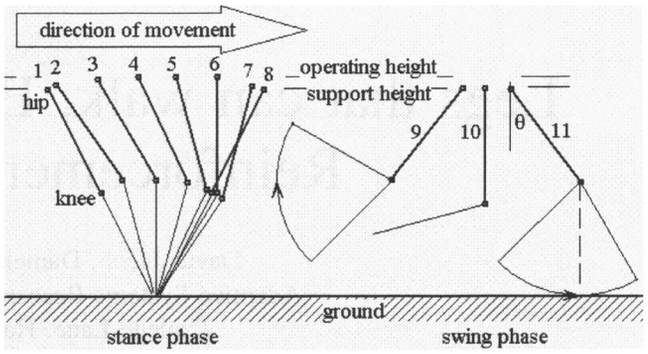

Fig. 2. A typical sequence of side elevations of the leg for one complete stride. See text for details.

the hip joint and also imposes large vertical forces which are likely to cause excessive movement in the supporting body: if too high, the foot will only be in contact with the ground for a small arc of the hip's motion, giving an inefficient overall action.

No additional compliance is provided in the system (other than that obtained as described above): high transient forces in the resulting rigid system therefore present a significant challenge to the learning algorithm.

For the present, the reciprocating motion of the hip is achieved with a ramp generator and is not learned. Many instances of such 'central pattern generators' (CPGs) are known in nature, see for example [1] and are usually assumed in walker models, *e.g.* [15].

The actions which are to be learned relate to the speed setting for the knee joint. An action is selected at equal time intervals 32 times during a complete cycle of the leg. Figure 2 shows, in diagram form, a typical sequence of side elevations of the leg, illustrating how the knee action needs to be varied to achieve the desired outcome: keeping the foot stationary on the ground during stance. Referring to the figure, the sequence of events is as follows.

Stance: View 1 represents 'first strike' of the leg. Here the leg will typically be fully extended and straight. Immediately, the knee begins to bend, so that although in 2 and 3 the torque at the hip is attempting to turn the upper leg clockwise, the lower leg's clockwise motion is actually providing the motive power here. The torque exerted on the upper leg by the knee forces the upper leg to move counter-clockwise. By view 4, clockwise motion of the upper leg has started but the angle of the lower leg with the ground makes it increasingly likely that it will start to slip as it transfers the force from the upper leg (which is now providing the power) to the ground. To prevent this, from view 5 the lower leg now needs a counter-clockwise torque from the knee, tending to straighten the leg and keep it in ground contact. This continues through 6 and 7; view 8 represents the very last contact during the stance phase, when the leg will tend to be straight.

Swing: Immediately the foot has left the ground at the completion of stance, the lower leg should quickly swing clockwise to be clear of the ground, as shown by the arc in view 9. View 10 shows an intermediate position of the leg as it swings forward (the maximum angle of bend at the knee is restricted to 1.5 radians, the angle shown) and 11 shows (as θ) the absolute minimum forward swing angle of the upper leg to allow the lower leg to swing into position for the start of the stance phase (the dashed line shows the point of minimum clearance). Larger θ will increase the clearance between the leg as it swings forward, and the ground.

Actions: Nine actions were available to the agent to set the desired rotational speed of the knee to different values. The number of actions and their values were chosen to give both slow movement suitable for fine adjustment during stance, and the fast movements required in the swing phase. All actions were available to the system at all times. A zero-speed option was one of those available. During early experiments several different partitions of the action range were tried; the one giving best results, comprising a set of nine possible actions, was used for further development.

III. Training the Leg

The difficulty this system is designed to overcome is simply stated: in order to learn to walk, a legged robot must first become sufficiently stable that it can take a few steps; once there is sufficient, stable support, the legs will quickly improve their performance. The question then is how this initial learning is to be accomplished, since the direct approach does not work.

The method adopted here is to construct a trainer, a simple mechanism which offers loading and dynamic characteristics similar to those which the leg will encounter in the robot, but in a context which is stable from the outset of learning. The design (figure 3) comprises a cantilever bar to support the leg: the fixed end of the cantilever is attached to a vertical axle (on the right of the figure) which is free to rotate. The cantilever is hinged across the mid-point of its length so that its outer half, to which the leg is attached, can swing up and down freely. A stop on this hinge prevents the attachment point of

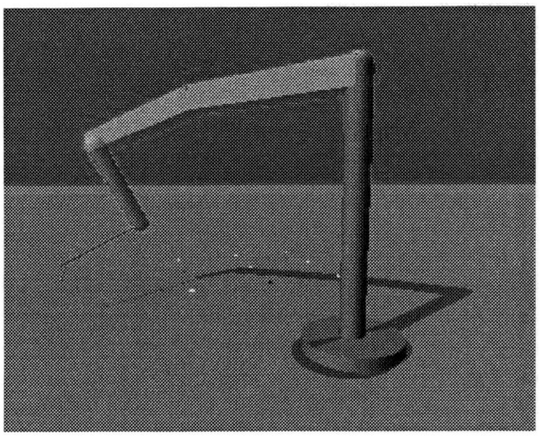

Fig. 3. The leg trainer

the leg falling below a certain height; this provides support for the leg during its swing phase.

The outer part of the cantilever has the same mass as the leg will be required to support in the walker; the momentum of this mass also continues the forward motion from the stance phase into the swing phase, but a degree of friction in the axle prevents the speed becoming excessive in operation; too great a speed would not reflect the expected behaviour of the integrated system.

Use of this training system is very simple: the leg's CPG is started, which initiates reciprocating motion in the upper leg. The lower leg performs initially random actions and is rewarded in accordance with the reward function described in section VI-A. As the leg learns, its stepping action causes rotation of the cantilever about the axle. During the training phase, the steps become more regular and assume a more even length.

A more intensive training régime was also tried, to simulate the effect of a less stable walking platform: here, the cantilever hinge's stop is movable to allow the leg to learn to operate at different support heights. The height was varied from the minimum clearance height for the leg up to the length of the vertical straight leg – the maximum height the leg can reach while still touching the ground. The variation was random and followed a gaussian distribution over this range.

In this latter case, learning was a little slower, and the average step length after learning was not so high, although it is hard to say whether this is simply the result of the fact that the leg was operating for part of the time in a less mechanically-efficient region. It is also not yet clear that this type of training actually confers advantages over the simpler one in the context of the walking robot - we return to this in section VIII.

IV. System Integration

Having trained the legs to operate in isolation, they are attached to a torso as shown in figure 4. Here again the mechanical design has to be considered.

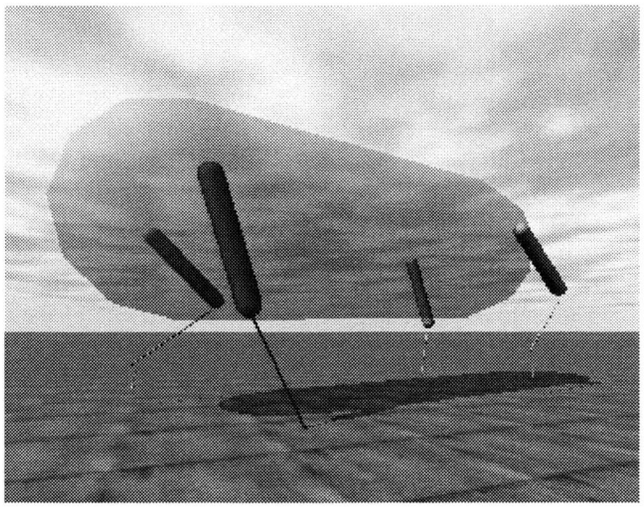

Fig. 4. The walker. The body is shown semitransparent so that all four legs can be seen

A. Mechanical Design

The legs are attached to either side of the extreme front and back of a cylindrical body with uniform mass distribution. With the current configuration of the system, where a CPG controls the legs, it is necessary to decide on a gait (the temporal sequence of footfalls) to be adopted.

There are many possibilities for different gaits in quadrupedal walkers and these have been extensively studied in relation to the horse and other animals (for an introduction and short bibliography see [6]). Nearly all gaits are 'dynamic', which is to say that balance can only be maintained during motion: the centre of mass is not always over the foot support area[2]. Any gait where only two or fewer feet are on the ground at any point, and where the feet have negligible ground contact area, as here, is therefore dynamic. In addition, gaits may or may not exhibit 'suspension', when all four legs are off the ground. The normal walk of a dog or a horse (where the legs are moved one at a time in alternating diagonal pairs) is not a suspended gait, whereas trot (two legs at a time are moved in diagonal pairs) and gallop (all four legs strike the ground in quick succession followed by a long period of suspension) are. Suspended gaits are alternatively called 'running' gaits.

Because of its symmetry and intrinsic stability, it was decided to employ a gait similar to trot but without suspension – instead, for a brief period in each cycle, all four legs are potentially simultaneously on the ground.

The mass of the body, which is being supported most of the time by two legs, is double the mass of the section of the cantilever arm which one leg has to support on its own during training.

Although walking was possible with the system as described, stability was enhanced by employing an additional mechanism which acts in a similar way to the pelvis and

[2]One of the rare exceptions to this in nature is the slow crawl performed by human infants, which is always statically balanced over three support points.

shoulders of a quadruped animal: the body twists in synchrony with the movement of the legs so that the non-supporting legs are raised further from the ground. The effect, which is achieved by cutting the torso in half along its midpoint and varying the speed of a rotary motor joining the two halves axially, can be adjusted by altering the phase (relative to the CPG), and amplitude of the twist.

It was found that the system is quite sensitive to some of the parameters and dimensions chosen for the walker. This appears to be largely due to the fixed rate of the CPG: depending on the motion of the body, leg strokes may occur too early or too late to perform their support function, with unpredictable results. Section VIII discusses this further.

The current system trains the leg with a support ratio of 3:1 (stance:swing) which is not suitable for a suspended gait. In fact this ratio does not occur in nature where the ratio at walk tends to be more of the order of 3:2 giving a more dynamic gait. The ratio used was chosen out of considerations of caution since it was felt that it might be somewhat ambitious initially to attempt too dynamic a gait (most legged robots have historically avoided dynamic gaits altogether, often by using six legs (*e.g.* [9]) and the alternating tripod gait, as insects normally do).

Despite the difficulties in maintaining dynamic gaits in robots, they have many advantages in terms of speed, manoeuvrability, smoothness and energy efficiency [3]. By transferring the weight from leg to leg, it becomes possible for the feet to 'grip' the ground rather than to slide ineffectually on the surface, as frequently occurs in mechanical walkers. However, highly responsive reactive control is required if this is to be realised.

V. EMBER

'EMbodiment-BasEd modulaR reinforcement learning' is the novel learning framework which we have developed during the investigation of which this work forms a part. Here we recapitulate the system by briefly explaining each of the terms which make up its name.

'Embodiment-based': Two aspects are important here. First, EMBER is designed for learning in embodied agents, that is to say, physical agents acting in the real world, or models thereof (hereafter 'real agents'). This already has important implications for learning: whereas classical RL is a general learning paradigm in which no prior assumptions are made, the behaviour of real agents is subject to natural restrictions, due to the intrinsic ordered structure of the physical world, which we can exploit as a source of constraints in the learning process. In this context, constraints are helpful: they restrict the range of possibilities which need to be considered at every step and so assist in guiding learning.

'Modular': The idea here is that, in addition, the particularities of the embodiment of a real system suggest a decomposition of learning corresponding to its physical structure. If we take biological agents as examples, a 'natural' modularity is often apparent in repetitive structures, multiple legs, say, or arms. Such structures can often themselves be conceptually decomposed into component parts, for example the joints of an arm or leg.

EMBER configures sensors and actuators within the real agent into modules which sense, act and generate reward locally; modules themselves are thus able to learn from their actions. The configuration of these modules reflects the physical structure of the embodiment on which they are based.

Modular decomposition of this kind is possible to the extent that we can assume that locally-similar states require locally-similar actions. If we can combine this local perception and action with a global, task-based reinforcement function, we will have systems which can learn about the effects of their actions in the world generally, at the same time as they learn a particular task.

The modular decomposition of a reinforcement learning problem also has important implications for its learning efficiency, owing to the factorisation of the learning space which it induces.

'Reinforcement Learning': EMBER is modular and makes use of multiple sources of reinforcement. These reinforcement signals must be combined to provide an overall action recommendation for the agent. EMBER incorporates novel mechanisms to achieve this, differing from previous methods in that the combination occurs before action selection. The agent therefore need not try actions it suspects to be bad just to establish the fact - it can simply avoid them. This ability to generalise is particularly beneficial to a real-world agent which might otherwise cause damage to itself or its environment by inappropriate actions.

The system described here in its current form does not yet feature global control, and does not make use of the algorithms we have so far developed to allow this. It does, however, demonstrate the both the EMBER principle that local actions based on local observation and learning can achieve a difficult, real-world task, and some of the advantages of the state-space factorisation which this modular approach affords.

VI. Learning

The choice of suitable state indicators and reward function is central to the design of a system of this kind. We aim to have learning proceed at a rate which is realistic in the context of real embodied systems, where learning times may be lengthened by orders of magnitude over their counterparts in simulation. Apart from the sheer size of a state-action space, one other factor greatly influences learning rate: the ability of the system to assign reward directly to the actions which have caused it. In many systems, particularly physical agents having mass (and therefore momentum) the effect of individual actions may still be felt several time steps later; thus a whole sequence of actions may contribute to an observed outcome, but it may be difficult to determine the extent of the contribution of each to the overall effect.

A classical RL approach to this training might be to apportion reward to the leg on the basis of velocity achieved, since rewards are conventionally derived from outcomes. However,

in each cycle of the leg, 32 actions take place each of which may or may not affect this, and the leg and its component parts have momenta which similarly spread the effect of individual actions. As each action is selected from a repertoire of nine, the number of possible action sequences over a single cycle is $32^9 = 3.5 \times 10^{13}$, any of which could potentially appear in a training run. A general reward given on the basis of one sequence may not tell us much about a different sequence, however similar; learning is likely to be very slow if it is possible at all. (In classical RL, given a very large number of training cycles and a stationary world, reward assignment will of course eventually be correctly made.)

An EMBER system has a further requirement: the local observability of variables both for the assignment of reward and for the determination of state. This self-imposed constraint has a practical benefit too: we may not know the nature of the system into which the legs will be integrated after initial training and cannot therefore rely on information being available other than what a leg can itself observe.

A. Reward Function

The main reward during the stance phase is given for avoiding slip between foot and ground. This will ensure forward motion (owing to the mechanical design of the leg) and keep the foot in contact with the ground so it is able to provide support to the body. However, during swing phase, this is precisely what we do *not* want; if the foot strikes the ground as the leg swings into place ready for the next stance, it is likely that the backward momentum it imparts to the leg will more than cancel out the forward motion achieved during stance. This is because the rotation of the hip joint is faster during swing than stance; as already noted, we want the stance phase to be longer than the swing phase to provide support for more than half the time.

To achieve this dual outcome, we need to have separate reward functions for the two phases, stance and swing, and synchronise the switching between them with the movement of the hip joint generated by the CPG.

For stability in any walker we will also want to limit excessive vertical forces which may arise, as we have seen, from the rigidity of the leg. Large forces may arise from an inappropriate action sequence during stance, or if the leg strikes the ground during swing; both should be avoided. Note that in both cases it is individual actions or possibly very short sequences of actions which will give immediate rise to these forces, so rewards can be assigned correctly: for this reason, learning to avoid them is practicable.

Finally, a walker will be more stable if its legs support it to equal heights, so a reward is given after each action proportional to the difference between the desired operating height and the actual height achieved. The operating height (which is relevant only during stance phase) does of course depend on a sequence of actions and the system's momentum will also affect it: in addition, the mechanical design of the leg is not conducive to maintaining constant height throughout its stroke. However, if the leg is in contact with the ground and forces are not excessive, the range of individual actions at each frame is quite constrained; in this case there may be a correlation between individual actions and the support height, and the results appear to suggest that an element of reward based on this parameter is effective in helping achieve the height desired, although it takes longer to learn than the forward motion, for instance.

B. State Indicators

The choice of variables to indicate state is crucial to the success of any reinforcement learning scheme. The designer must decide how many features to observe, what they should be, and (in the case of continuous variables, as here) how to partition the observable range into discrete values. The aim is to capture the most indicative aspects of the system's behaviour in the smallest possible state space. Possibilities in this case include but are not limited to the position, angular speed or acceleration of each of the two joints; the magnitude or direction of the force transmitted by the leg; and the angle of contact of the leg with the ground. Since the hip joint is operated by a ramp generator cycling through a time series of states, these states themselves could be indexed and used as a state indicator. And the leg will also need to know, since the reward functions are different, whether it is in stance or swing phase. This will require a dedicated indicator bit unless it is implicit in other indicators (e.g. a time series index).

Although various methods have been proposed to automate the selection of features, for example McCallum's U-trees [14], and state-frequency analysis or similar methods can be used to determine the effectiveness of a given partition of each, these are unlikely to help us here for the following reasons. First, such methods require *all* possible features to be used initially, each with an arbitrary partitioning. But the key difficulty here is that the importance of individual features can only be determined as the system learns, which given the likely size of the initial state-space might take a very long time to achieve. Further, in this particular example, owing to the need to determine the phase, the absence of some possible features might make learning impossible; again, arbitrary partitioning of features may miss crucial distinctions without which learning could not take place, or introduce a large number of functionally identical states which initially, at least, could not learn from one another.

Accordingly, using domain knowledge available to the designer, a number of likely schemes was devised and tried experimentally and the most efficient in terms of learning speed and outcome chosen for further development. In this scheme the variables monitored were:

- hip joint angular position (17 partitions)
- vertical force through leg (8 partitions)
- angle between foot and ground (8 partitions)
- stance or swing (1 bit)
- hair sensor (1 bit)

The last item is a simple sensor in the form of a lightweight deformable hair attached to the front of the leg near the ground (figure 1). This was provided to give an indication,

during the swing phase, that the leg was very close to the ground: this information was not available from the other features. Similar hair-like sensors, performing a variety of sensing functions, are found on insect legs. The sensor reads 1 if the hair is in contact with the ground, 0 otherwise.

C. Learning algorithm

The learning algorithm employed is single-step Q-learning [23]. A comprehensive treatment of RL principles and techniques including Q-learning is provided by [22]. The simplicity and generality of this mathematically-sound and well-understood algorithm appear well-suited to this particular task, although the EMBER framework does not require the use of any particular algorithm in this case.

The legs are identical in form and function: following the EMBER principle that, in general, similar local observations require similar local actions, only one leg need be trained. When several legs are subsequently used in the integrated system, they can therefore all reference the same instance of the state-action-reward table and update it as they learn: as a result, learning after integration will happen four times as quickly. It might of course be the case that the legs may need to specialise to an extent when in situ on the body. This may be due to symmetric reversal, in which case sensors and actuators can be reconfigured to suit; otherwise, the use of more than one learning space may give better results, at the cost of longer learning times.

This sharing of the learning space does not, of course, mean that the legs perform the same actions as one another, even when they act simultaneously; in general, two legs perform the same action only when that is the optimum action for each leg's individually-determined state. The legs therefore are able to act entirely independently but according to the same learned pattern. Such independent action is essential in the context of a dynamic system such as that described, since it enables a reactive response to changing conditions.

VII. RESULTS

Because of the large number of parameters involved and the complexity of the system overall, it is impractical to attempt a detailed description of the experiments carried out. Similarly, the specimen results here are intended to give an overview of the performance obtained.

A. Trainer

The leg was trained for 6000 steps before incorporation in the walker. Figure 5 shows the effect on learning of support height modulation. Walker results were obtained using the leg trained without height modulation.

B. Walker

The success of experiments of this kind is hard to assess quantitatively. There is the crude measure of 'mean steps to failure' which is suitable where systems fail frequently - in this case, failure occurred at 62 steps, and again at 5473 steps after integration. Otherwise, the measure which perhaps best

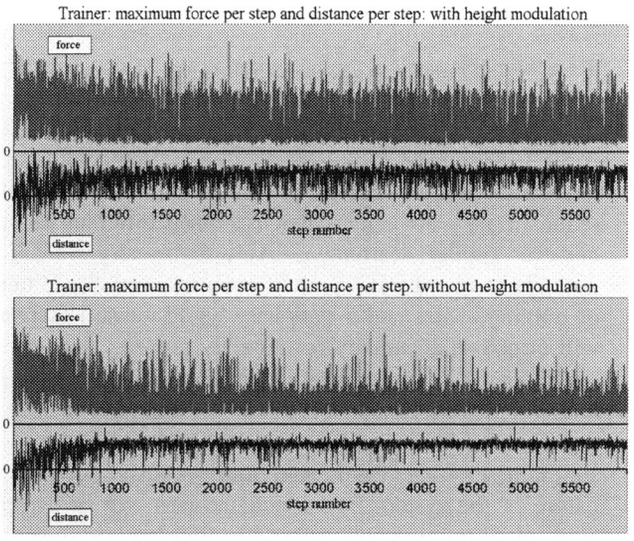

Fig. 5. Trainer: maximum force and distance travelled per step, for a training period of 6000 steps. Upper graph shows results from the more intensive training régime, where the leg support height was varied by up to 10%. In the lower graph the support height was constant.

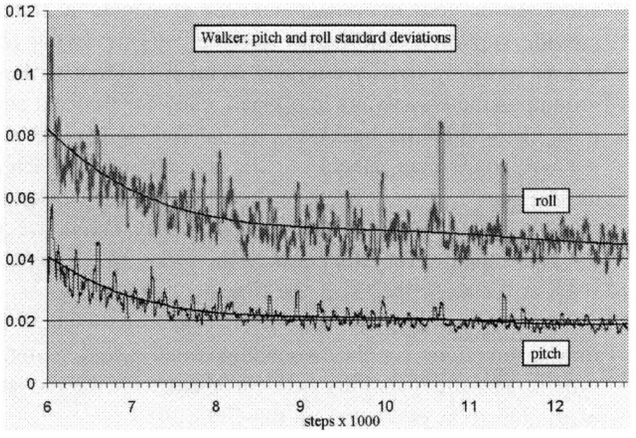

Fig. 6. Standard deviation of roll and pitch amplitudes against number of steps taken, with 4th order polynomial trend lines

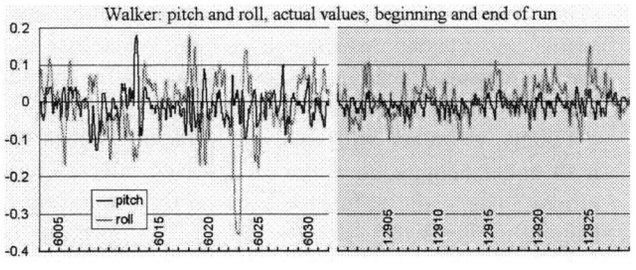

Fig. 7. Actual measurements of pitch and roll (in radians deviation from the horizontal) for representative samples of steps at the start of integration and after 6900 steps of learning.

expresses an observer's qualitative impressions of the walker's stability is the extent and regularity of the body's pitch and roll motion. Figure 6 clearly shows that learning continues to improve general performance for several thousand steps after integration; figure 7 shows the actual pitch and roll at the beginning of the run and towards the end. Here the 'beating' effect of the difference between the natural frequency of the structure and the step rate can clearly be seen.

VIII. Discussion

One of the criticisms often levelled at Reinforcement Learning (*e.g.* by Brooks [4]) is that, although mathematically elegant, it is impractical because first, it requires the Markov condition to be met (which is rarely the case in the physical domain with a limited number of sensors) and second, learning times are too long for real robot systems. The work we describe goes some way to showing that RL is capable of useful learning even in processes which are demonstrably not strictly Markovian: the EMBER principle that *similar* local states require *similar* local actions suggests that sub-optimal actions may be substituted provided they are not too dissimilar[3] to the optimum action, and learning will still take place. The ability of the leg to learn some control even where its support height was modulated during training shows this quite strongly. The second, learning time issue is tackled both by the reduction in state-space size due to the factorisation effect of the modular decomposition, and also due to the immediacy (both temporal and spatial) of connection of action and reward within the modules. Naturally there is a trade-off to be made – this system cannot be guaranteed to learn optimum solutions, but as in the experiments reported here, a sub-optimal solution may serve us well if there is no better alternative achievable in practice.

The main source of systematic instability in the walker occurs because the whole structure behaves as an inverted pendulum with its own resonant frequency which does not coincide with the frequency of the CPG. The result of this is that the weight carried by each leg, and its operating height as the torso swings, vary unpredictably. The response to this has to be learned in situ: the trainer does not simulate it. However, as the walker learns, it becomes more able to cope with this, either by the legs altering their effective compliance and therefore the resonant frequency of the structure, or by becoming more resistant to the effects of these perturbations, or both. During the experiments, though, there always remained a small probability of failure, even after extended learning, as in the specimen results included here. In future, more biologically-inspired oscillators capable of adaptive synchronisation *e.g.* [5] may be tried in place of the fixed-period model currently in use.

However, as mentioned in section IV, often this instability is too great to be overcome simply by further learning - the system is unable to dissipate the forces which arise from mechanical mismatch. In this connection, it has been established that the distribution of mass in the body has a fundamental influence on the behaviour of legged systems with dynamic gaits [17], and this accounts for some of the parameter sensitivity experienced. Notwithstanding our earlier remarks, the success of a RL system of this type does depend on its task being *approximately* Markovian with respect to its state - in this case the state is measured by local sensors and cannot take account of variations further afield. This demonstrates that mechanical considerations must always form a large part of the design process for a physical system; however good the learning algorithm, these will set the upper bound on its performance.

Further, in many cases, local observation cannot be substituted for global. In the current experiments, we use the slip of the leg on the ground as a substitute for measuring the global velocity. This is adequate, as we have shown; in other cases, however, there might be no suitable local heuristic for a particular global observation. It is not immediately apparent how we could assess directional stability in the same way, for instance.

It is known that in some insects the switch from swing to stance is initiated by pressure on the tarsus, the tip of the leg, and this is also used in [7]. If implemented in our system, this would almost certainly result in a considerable improvement in performance. However, if we are to learn, rather than program, the swing phase, it remains a question how this would be achieved. The walker points up an interesting problem with the use of reinforcement learning techniques in this context: although it is clear to a human observer what the leg should do during the swing phase, that is, withdraw as far as possible from the ground, it is difficult to train this behaviour since the leg usually avoids hitting the ground even when it fails to retract fully. The long intervals between punishment therefore result in slow learning of this behaviour, but because it represents a major source of potential instability it is important that it should be learned.

One thing which became apparent is that it is better to train for the behaviour you want to achieve rather than try to train for situations which might be encountered. In the current experiments, this is reflected in the fact that the simpler training régime gave more stable results in the walker, and subsequent learning based on this was faster and more effective.

IX. Related Work

There are indications that in some insects, notably the cockroach, the action of the legs is largely autonomous and independent of the animal's central nervous system, and a simplified version of its neural circuitry has been incorporated into hexapod robots able to negotiate rough terrain and explore their environment [7], [9]. Force as a determinant of state is important in the above models, is used in our model, and is the subject of in-vivo cockroach experiments [1].

Reinforcement Learning has been used in a variety of walking systems, ranging from the complex, mechanical biped

[3]because we consider exclusively systems acting in the physical world, the 'similarity' of actions is a meaningful concept

system of [2] to the CPG-actor-critic model of [15], and in automated gait development in AIBO robots [12] but in all of these cases the actions consist in modifying some parameters of an existing gait or gait generation system, rather than using action primitives as in our work.

Hierarchical reinforcement learning is the topic of much recent and ongoing research: the modular decomposition of a task into sub-tasks to improve efficiency by reducing learning redundancy is tackled for example in [8], [21], [18]. However, general principles underlying the selection of modules are difficult to identify analytically, although this often presents no difficulty for a human designer. Our EMBER framework is somewhat different: the modules are based on identifiable physical structures, with their own capacity for observation, action and learning.

X. Conclusions and Further Work

This paper introduces a number of new techniques:

1) a mechanical system which is too complex and whose component parts interact to such an extent that it cannot learn a task, is physically decomposed into modules which are trained separately. These modules can determine state and reward on the basis of local observation, that is, from sensors directly mounted on them, and act using their actuators. Initial training takes place in a simplified, stable analogue of the complete system, after which the modules are replaced on the body. The integrated system is now stable enough for learning to continue.
2) Several modules performing the same task share the same learning (state/action) space and learn simultaneously: this ensures symmetrical response and reduces learning time (clock time).
3) A task comprising two incompatible elements (the swing and stance phases of the leg motion) is learned by switching the reward function at the appropriate point in the cycle. A corresponding indicator bit in the learning space effectively divides it into two learning spaces corresponding to the two phases, which can thus be learned in tandem.

The work demonstrates the EMBER principle that similar local observations require similar local actions; it shows that the modular decomposition of complex embodied agents along lines suggested by their physical structure can be a useful approach in tackling hard problems in the physical domain.

However, there is clearly much to be done to make a useful, practical system on the basis of this model. There remains the question of central control: we need to modulate the response of the legs to do other things, like walk on a curve or at different speeds. Areas for possible investigation include a 'library' of different trained steps to select from, or perhaps a mechanism for interpolating actions between two or more trained examples.

More immediately, the use of tarsal pressure to initiate stance, leading to autonomous gait generation and, it is hoped, greater stability, will be investigated.

References

[1] Turgay Akay, Sebastian Haehn, Josef Schmitz, and Ansgar Büschges. Signals from load sensors underlie interjoint coordination during stepping movements of the stick insect leg. *Neurophysiology*, 92:42–51, 2003.

[2] H. Benbrahim and Judy Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, December 1997.

[3] Michael Brady, editor. *Robotics Science*, chapter 16, "Legged Robots", Marc H. Raibert. MIT Press, 1989.

[4] R. Brooks. From earwigs to humans. *Robotics and Autonomous Systems*, 20(2 - 4):291 – 304, June 1997.

[5] J. Buchli and A. J. Ijspeert. Distributed central pattern generator model for robotics application based on phase sensitivity analysis. In *The First International Workshop on Biologically-Inspired Approaches to Advanced Information Technology (Bio-ADIT 2004)*, 2004.

[6] Stephen Budiansky. *The Nature of Horses*. The Free Press, New York, 1997.

[7] H. Cruse, U. Mueller-Wilm, and J. Dean. Artificial neural nets for controlling a 6-legged walking system. In *From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 108–115, 1992.

[8] T. G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Artificial Intelligence Research*, 13:227 – 303, 2000.

[9] M. Frik, M. Guddat, D.C. Losch, and M. Karatas. Terrain adaptive control of the walking machine tarry ii. In *European Mechanics Colloquium, Euromech 375 - Biology and Technology of Walking*, pages 108–115, 1998.

[10] David Jacob, Daniel Polani, and Chrystopher L. Nehaniv. Ember: Learning with embodiment-based modular reinforcement. Technical Report 398, University of Hertfordshire, Faculty of Engineering and Information Sciences, Hatfield, UK, January 2004.

[11] David Jacob, Daniel Polani, and Chrystopher L. Nehaniv. Improving learning for embodied agents in dynamic environments by state factorisation. In *TAROS 2004, Towards Autonomous Robotic Systems*, September 2004.

[12] Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004.

[13] Pattie Maes and Rodney A. Brooks. Learning to coordinate behaviors. In *National Conference on Artificial Intelligence*, pages 796–802, 1990.

[14] Andrew K. McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, University of Rochester, New York, 1996.

[15] T Mori, Y Nakamura, Masa-aki Sato, and S. Ishii. Reinforcement learning for a cpg-driven biped robot. In *Nineteenth National Conference on Artificial Intelligence (AAAI)*, pages 623–630, 2004.

[16] J. Morimoto and K. Doya. Reinforcement learning of dynamic motor sequence: Learning to stand up. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1721 – 1726, 1998.

[17] K.N. Murphy and M.H. Raibert. Trotting and bounding in a planar two-legged model. In *Fifth Symposium on Theory and Practice of Robots and Manipulators*. MIT Press, 1985.

[18] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems*, volume 10, 1997.

[19] William D. Smart and Leslie P. Kaelbling. Effective reinforcement learning for mobile robots. In *International Conference on Robotics and Automation*, May 2002.

[20] Russell Smith. Open dynamics engine. WEB: www.ode.org, 2004.

[21] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[23] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.