7^{th} Int. Symposium on Robotics and Applications, Anchorage, AK

A CONTROL STRUCTURE FOR LEARNING LOCOMOTION GAITS *

MANFRED HUBER and RODERIC A. GRUPEN
Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts, Amherst MA 01003

ABSTRACT

Learning and adaptation are important for robot systems operating in the real world in order to react to changes in the task requirements. In many domains this involves the acquisition of cyclic behavioral patterns requiring repetitive control strategies. In the domain of legged locomotion in particular, walking gaits must be acquired. This paper presents a hybrid control architecture which allows to learn such cyclic control strategies while providing reactivity at the lowest level through a set of feedback controllers. The use of reinforcement learning on top of a Discrete Event Dynamic System (DEDS) model of the system behavior furthermore permits to learn such gaits in a single trial using a simple reinforcement structure while maintaining the safety of the mechanism through the imposition of safety constraints. To illustrate this approach it has been applied to the learning of a turning and a forward walking gait on a quadruped robot.

KEYWORDS: legged locomotion, reinforcement learning, DEDS

INTRODUCTION

Autonomous systems operating in an uncertain environment must be flexible to address a large number of tasks and a variety of run-time conditions. In addition such systems have to be reactive and adaptive to address new environmental conditions and to learn new tasks. This requires the control system to be able to exhibit a wide variety of behavior while computational pressures limit the complexity of the behavioral representation. This in turn leads to strategies that reuse constituent controllers in repetitive control sequences rather than advocating monolithic controllers.

A large body of work in legged locomotion has focused on constructing such cyclic walking gaits as geometrical sequences of foot placement locations [2, 7]. Such approaches, however, are very sensitive to imprecision in the terrain maps or the effects of control actions. In addition, most learning approaches to geometric foot sequences

*This work was supported in part by NSF IRI-9503687, IRI-9704530, and CDA-9703217

have to acquire gaits off-line in simulation since either complete supervision of the learning process is required or the system exhibits intolerable behavior in the learning phase. To address uncertainties in the terrain knowledge and make the system more reactive, behavior-based approaches have been used to augment preplanned gait sequences [4, 11] or to generate foot placements on-line [1] . Gaits of this kind have the advantage of being more flexible and permit on-line learning [8]. The character of the behaviors used, however, result in geometric gaits that are still rather inflexible.

To make learning approaches to legged locomotion useful in autonomous agents, learning techniques have to be able to acquire gaits on-line without extensive outside supervision. The approach presented here uses reinforcement learning on a control structure which forms locomotion gaits as sequences of controllers. The generic character of these controllers allows a large number of different gaits to be constructed. Furthermore, control knowledge in the form of resource schedules allows policies to encode multiple control solutions. These families of control policies permit flexible responses to locked resources or task interactions. The resulting gait is represented as a finite state machine controller and thus potentially contains a large number of control cycles, each applicable to a variety of run-time contexts.

In this paper, this approach is used to learn gaits for a four-legged walking robot on even terrain. In particular, turning and forward walking gaits are learned using a reinforcement signal encoding instantaneous rotational and translational progress.

THE CONTROL ARCHITECTURE

Figure 1 illustrates the control architecture proposed for learning gaits on-line.

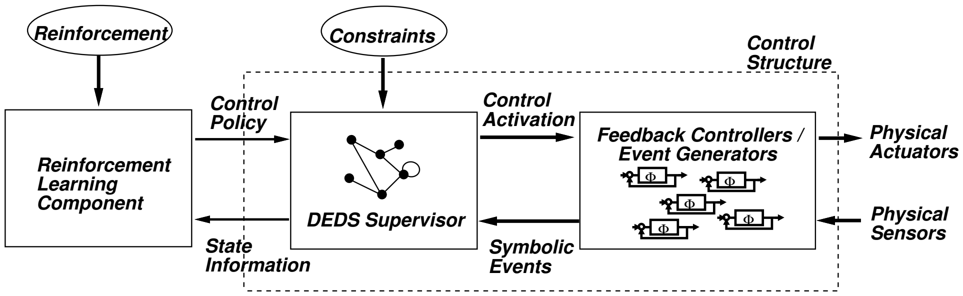

Figure 1. The control architecture

Here behavior is derived from a set of feedback control laws which can be bound dynamically to subsets of the system resources. Control policies within this framework take the form of sequences of concurrent controller activations. This and the generic character of the control laws permits a small number of these to span a potentially large number of tasks in a flexible fashion. Furthermore, the asymptotic stability of the constituent controllers used in the current implementation transforms the underlying continuous space into a set of discrete system equilibria. The state of the system can thus be characterized abstractly by means of predicates representing the functional goals of the individual controllers. Using this abstraction, system behavior can be modeled as a DEDS which forms the basic structure for a subsequent reinforcement learning problem. Formal techniques in the DEDS framework allow constraints to limit exploration to safe and relevant control alternatives. Using such a constrained DEDS supervisor which takes the form of a nondeterministic finite state machine, a reinforcement learning component learns the transition probabilities within the underlying model, as well as a control policy which optimizes the given reinforcement. Overall, this structure dramatically reduces the complexity of the learning problem by reducing the size of the state and action spaces. This in turn permits the acquisition of these policies without outside supervision in a single trial.

Control Basis

To address legged locomotion, the control basis used here consists of three control laws which address generic control objectives for locomotion and manipulation tasks:
Φ_0: **Configuration space motion control -** a harmonic function path controller is used to generate collision-free motion of the robot in configuration space.
Φ_1: **Contact configuration control -** contact controllers locally optimize the stability of the foot pattern based on the local terrain.
Φ_2: **Kinematic conditioning control -** a kinematic conditioning controller locally optimizes the posture of the legs.
Each of these basis control laws Φ_i can be bound on-line to input resources σ (sensors or sensor abstractions) and output resources τ (actuators) derived as subsets of the system resources (legs $0, 1, 2, 3$ and position and orientation of the center of mass x, y, φ) of the four-legged robot illustrated in Figures 2 and 3. The resulting

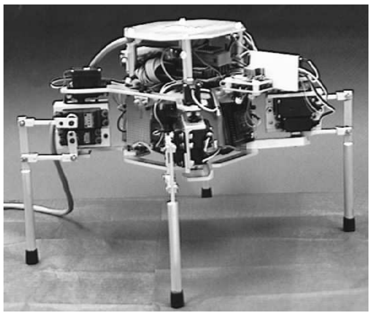

Figure 2. Walking robot

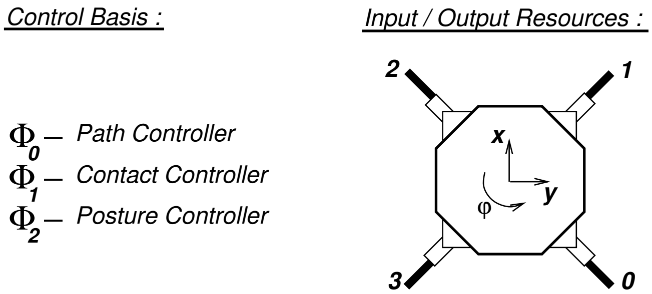

Figure 3. Controller and resource notation

feedback controllers $\Phi_{i}{}_{\frac{\sigma}{\tau}}$ can be activated concurrently according to a task dependent composition policy under the "subject to" ("$\triangleleft$") constraint. This constraint restricts the control actions of subordinate controllers such that they do not interact destructively with the objectives of higher priority controllers. The composite controller $\Phi_{2}{}_{\frac{0,1,2,3}{\varphi}} \triangleleft \Phi_{1}{}_{\frac{0,1,2}{0}}$, for example, attempts to achieve a stable stance on legs 0, 1, and 2 by moving leg 0 with the dominant controller while the subordinate controller optimizes the kinematic posture of all four legs within the "nullspace" of Φ_1 by rotating the body. A complete control policy takes the form of a sequence of such concurrent controller activations, where different task objectives are achieved by different composition policies rather than by designing new control elements. This control basis has already been used successfully for dextrous manipulation and locomotion tasks [3, 6] using hand-designed composition policies.

Control Structure

This set of control laws and robot resources then leads to a functional description of the state of the system in terms of the control objectives of the control laws. For the locomotion examples in this paper, a maximum of two controllers are allowed concurrently and the set of controllers is limited by removing the path controller. This reduces the predicate space to the predicates $(p_1, p_2, p_3, p_4, p_5,\ p_6, p_7, p_8, p_9, p_{10})$ corresponding to controller/input binding pairs in the following way:

$$\begin{array}{lllll} p_1 \leftarrow \Phi_1{}^{1,2,3}_{\underline{*}} & ,\ p_2 \leftarrow \Phi_1{}^{0,2,3}_{\underline{*}} & ,\ p_3 \leftarrow \Phi_1{}^{0,1,3}_{\underline{*}} & ,\ p_4 \leftarrow \Phi_1{}^{0,1,2}_{\underline{*}} & ,\ p_5 \leftarrow \Phi_1{}^{0,1,2,3}_{\underline{*}}\ , \\ p_6 \leftarrow \Phi_2{}^{0}_{\underline{*}} & ,\ p_7 \leftarrow \Phi_2{}^{1}_{\underline{*}} & ,\ p_8 \leftarrow \Phi_2{}^{2}_{\underline{*}} & ,\ p_9 \leftarrow \Phi_2{}^{3}_{\underline{*}} & ,\ p_{10} \leftarrow \Phi_2{}^{0,1,2,3}_{\underline{*}}\ , \end{array}$$

where $\underline{*}$ indicates the independence of the predicate evaluation from the output resource. $p_1 - p_5$ indicate thus the existence of stable three and four-legged stances, while the last 5 predicates represent the favorable posture of the legs.

In the predicate space, the system is then modeled as a DEDS where convergence of the controllers leads to transitions between states. Using this framework, it is possible to impose constraints on the behavior of the system [5]. In the examples here, a safety constraint for quasistatic walking is imposed, requiring that at least one stance has to be stable at all times. In terms of the predicates this implies that $p_1 \vee p_2 \vee p_3 \vee p_4 \vee p_5$ must always be true. In addition, knowledge about the platform can be introduced in the form of domain constraints to reduce the complexity of the system model. In a quadruped robot, for example, a stable three-legged stance automatically implies a stable four-legged stance. Furthermore, for the platform employed here, kinematic limitations do not allow the simultaneous stability of two opposing support triangles. These pieces of information can be added by means of the constraints $\neg(p_1 \vee p_2 \vee p_3 \vee p_4) \vee p_5$ and $\neg(p_1 \wedge p_3) \wedge \neg(p_2 \wedge p_4)$, respectively. Throughout the following learning process, actions are then limited by the DEDS supervisor to conform with the constraints, thus ensuring the stability of the platform.

Reinforcement Learning Component

To address new tasks in the system, a learning component is used to acquire a control policy on top of the DEDS supervisor. In an autonomous system, reinforcement learning provides a suitable mechanism since this paradigm is exploration-based and can learn from a simple reinforcement signal. Furthermore, this scheme allows the acquisition of control policies whose objective is not represented as a state in the underlying state space and thus permits cyclic control policies. In the experiments presented here, Q-learning [10] is used to learn locomotion gaits for a given reinforcement signal. At the same time, the exploration is used to estimate transition probabilities between predicate states and thus to improve the abstract model of the system behavior. While these probabilities are not used explicitly in the experiments, they could be used for off-line learning on this approximate system model [9].

EXPERIMENTS

To show its applicability, this approach is applied to two different locomotion tasks, a purely rotational gait and a forward walking gait. In both cases the system is started in a stable stance and possible actions are selected using a Boltzmann distribution where the temperature is slowly decreased throughout the learning process. Learning occurs in the course of a single trial without the need to reset the robot. Due to the safety constraints imposed in the DEDS layer, the system always maintains a stable stance, making supervision by a human unnecessary.

Different locomotion gaits are acquired by means of varying the reinforcement function. In the case of the turning gait, a reinforcement structure is used which rewards the system for performing rotational progress and punishes it for translational displacements. The resulting learning curve is shown in Figure 4 where exploration is turned off after $20,000$ learning steps.

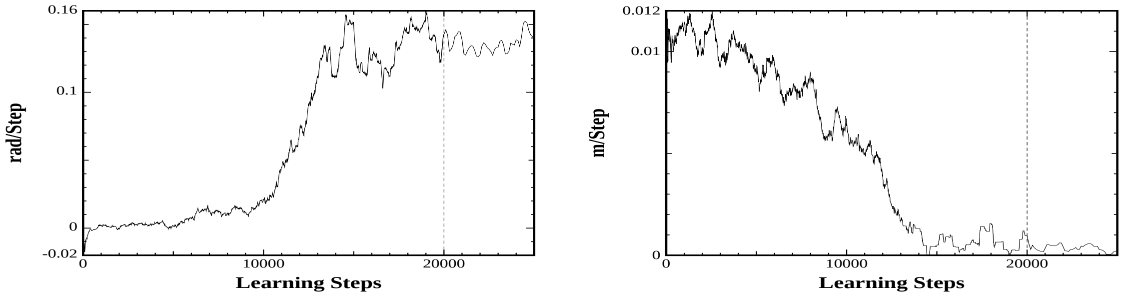

Figure 4. Rotational progress (left) and magnitude of associated translation error (right) throughout learning of the rotation task

These two graphs show the rotational progress and the magnitude of the translational displacement per control step, respectively. The system learns to achieve reliable rotation while simultaneously reducing the amount of translation. The learning process finally results in a finite state controller which reliably cycles through a succession of stable three legged stances such that a continuous rotation is achieved.

In a similar way, the translation gait is learned by means of a different reinforcement structure. In this case, the system is rewarded for instantaneous forward progress and punished slightly for any body rotation performed. Again, the graphs in Figure 5 show that the system is able to learn a gait pattern which achieves translational progress while reducing the amount of rotation. It is important to notice here, however, that

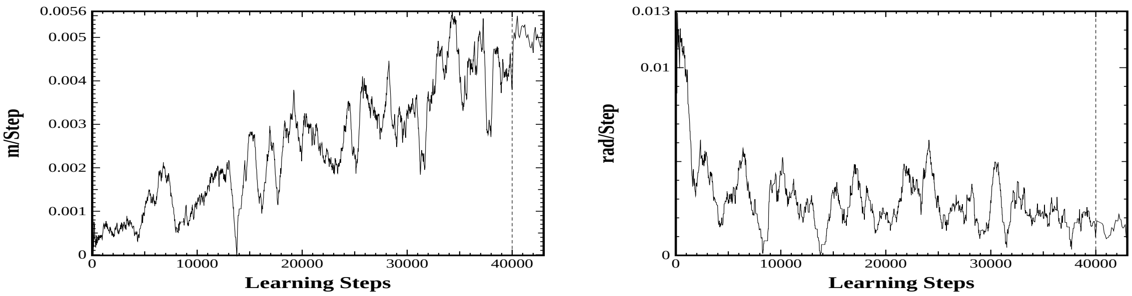

Figure 5. Forward progress (left) and magnitude of associated rotation (right) throughout learning of the translation task

even without any exploration after learning step $40,000$, the gait still contains some body rotation. Furthermore, the finite state controller contains multiple cyclic control strategies which are executed depending on the initial conditions. Throughout execution, the system switches between different such cycles while constantly achieving forward progress.

While the results presented here were obtained using a simulator of the robot, similar experiments to learn a rotation gait with a further restricted set of control alternatives were also performed successfully on-line on the real robot.

CONCLUSIONS

Learning and adaptation are important for robot systems operating in an uncertain environment. Such systems have to be able to react to new conditions and task requirements and to learn new behavioral policies without the need for outside supervision whenever the task requirement change. For many task domains this involves the acquisition of cyclic behavioral patterns requiring repetitive control strategies. In the domain of legged locomotion in particular, walking gaits typically involve roughly periodic sequences of leg movements. To address these requirements, the control architecture presented here derives behavior from a set of continuous control modules which are coordinated in an abstract DEDS model of the system behavior. This model allows the imposition of safety constraints on the system behavior, thus permitting exploration to be performed without encountering catastrophic failures. Walking gaits are then learned using reinforcement learning on top of the DEDS structure, resulting in finite state controllers for various walking objectives.

REFERENCES

1. R. A. Brooks. A robot that walks; emergent behaviors from a carefully evolved network. *Neural Computation*, 1(2):355–363, 1989.
2. P. de Santos and M. Jimenez. Generation of discontinuous gaits for quadruped walking vehicles. *J. Robotic Sys.*, 12(9):599–611, 1995.
3. R. A. Grupen, M. Huber, J. A. Coelho Jr., and K. Souccar. Distributed control representation for manipulation tasks. *IEEE Expert*, 10(2):9–14, April 1995.
4. S. Hirose. A study of design and control of a quadruped walking vehicle. *Int. J. Robotics Res.*, 3(2):113–133, 1984.
5. M. Huber and R. A. Grupen. A hybrid discrete event dynamic systems approach to robot control. Technical Report 96-43, CMPSCI Dept., Univ. of Mass., Amherst, October 1996.
6. M. Huber, W. S. MacDonald, and R. A. Grupen. A control basis for multilegged walking. In *Proc. Int. Conf. Robot. Automat.*, pages Vol.4 2988–2993, Minneapolis, MN, April 1996. IEEE.
7. K. Jeong, T. Yang, and J. Oh. A study on the support pattern of a quadruped walking robot for aperiodic motion. In *Proc. IROS*, pages 308–313, Pittsburgh, PA, August 1995. IEEE.
8. P. Maes and R. Brooks. Learning to coordinate behaviors. In *Proceedings of the 1990 AAAI Conference on Artificial Intelligence*. AAAI, 1990.
9. A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, 13, 1993.
10. C. J. C. H. Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, Cambridge, England, 1989.
11. D. Wettergreen, H. Pangels, and J. Bares. Behavior-based gait execution for the dante ii walking robot. In *Proc. IROS*, pages 274–279, Pittsburgh, PA, August 1995. IEEE.