BioSystems 112 (2013) 102–106

Contents lists available at SciVerse ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Evolving locomotion for a 12-DOF quadruped robot in simulated environments

Gordon Klaus*, Kyrre Glette, Mats Høvin

University of Oslo, Department of Informatics, Ole-Johan Dahls Hus, Gaustadallen 23 B, N-0373 Oslo, Norway

ARTICLE INFO

Keywords:
Evolutionary robotics
Simulation
Locomotion

ABSTRACT

We demonstrate the power of evolutionary robotics (ER) by comparing to a more traditional approach its performance and cost on the task of simulated robot locomotion. A novel quadruped robot is introduced, the legs of which – each having three non-coplanar degrees of freedom – are very maneuverable. Using a simplistic control architecture and a physics simulation of the robot, gaits are designed both by hand and using a highly parallel evolutionary algorithm (EA). It is found that the EA produces, in a small fraction of the time that takes to design by hand, gaits that travel at two to four times the speed of the hand-designed one. The flexibility of this approach is demonstrated by applying it across a range of differently configured simulators.

1. Introduction

Robot design can be complex. To design by hand, one often needs skills and experience with mechanics in general and with the particular robot and its dynamics. The space of possible solutions is enormous, so, to make the problem tractable, the designer is forced to make simplifying assumptions, for example by dividing the problem into more manageable pieces and limiting the dependencies among them; furthermore, one is typically biased towards modern engineering conventions and personal experience. As a result, many potential solutions are not even considered. Thus, one cannot in general expect to design an especially good, let alone optimal, robot by traditional methods – perhaps not even with a very significant investment of effort.

Evolutionary robotics (ER) (Doncieux et al., 2009; Harvey et al., 1997) solves this problem, to some extent. When using ER, the necessary problem-specific competencies consist only of what is required to construct and evaluate the robot. One must still define, and thus limit, the solution space, but this can be done in a manner unmotivated by the amenability of the resulting problem to manual analysis; bias can thus be freely controlled and nontraditional architectures explored. ER permits the designer to focus more energy on high level goals – i.e., on defining a fitness function – and it allows him to explore the vast space of possible solutions, including complex designs otherwise unlikely to be considered: for example, tensegrity structures and soft muscle-like actuators (Rieffel et al., 2008, 2009; Glette and Hovin, 2010). Whole families of robot architectures can be explored by allowing morphological parameters to vary within the optimization process (Bongard, 2003; Macinnes and Di Paolo, 2004; Rieffel et al., 2009).

While in its barest form ER involves evaluating candidate solutions in the real world, there is an enormous benefit to be gained by using a simulator. With sufficient computing power, many individuals can be evaluated in simulation in the time it would take to evaluate one in reality, while at the same time eliminating mechanical wear and unreliable human intervention. But there is no free lunch: in order to achieve this speedup, the simulator must sacrifice some accuracy; this yields a disparity between the simulated and real behaviors which has been termed the "reality gap" (Jakobi et al., 1995). Some efforts are currently underway to deal with this issue (Koos et al., 2010; Bongard et al., 2006). The other outstanding issue in ER (and with EAs in general) is the difficulty with scaling it to very complex designs while maintaining a reasonable time to convergence. Both of these issues are discussed in Doncieux et al. (2009).

ER is far from perfect but it shows a great deal of promise. In this paper, we just begin to scratch the surface of its capabilities, laying a foundation for more in-depth investigations. We aim to demonstrate the effectiveness of ER by using it to design control systems for the simulated locomotion of a novel quadruped robot on flat ground.

In the next section we introduce the robot, the simulator, the EA, and the hand-designed gait. Thereafter, the results are discussed. Finally, we conclude the paper and describe our plans for future work.

2. Implementation

The robot that is the focus of this paper is a quadruped that has been designed in our lab and which is currently being constructed.

* Corresponding author.
E-mail addresses: gordonk@ifi.uio.no (G. Klaus), kyrrehg@ifi.uio.no (K. Glette), matsh@ifi.uio.no (M. Høvin).

0303-2647/$ – see front matter © 2013 Elsevier Ireland Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.biosystems.2013.03.008

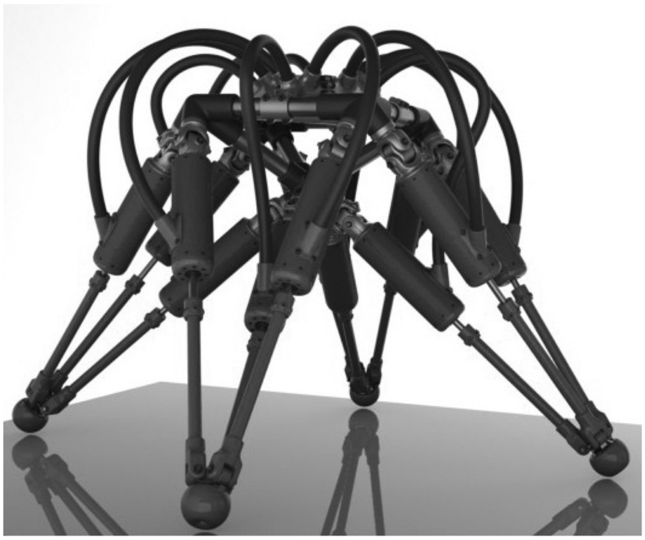

Fig. 1. A rendering of the robot model.

As can be seen from the rendering in Fig. 1, its body is an inverted rectangular pyramid and each of its legs consists of three linear actuators meeting at a spherical foot. The three non-coplanar degrees of freedom in each leg allow the foot to move freely in a large volume of space, providing the strength and wide range of motion that will be requisite for its ultimate role as a climbing robot; this should also make it an adept walker.

To simulate the robot's motions, the PhysX (NVIDIA, 2013) physics simulation software library was used. A model of the robot, shown in Fig. 2, was created in PhysX to capture the salient aspects of its design such as the types and connectivity of its joints and its rough shape. The ultimate goal is, of course, to make the simulation as realistic as possible so that designs can be transferred into the real robot, when it exists. Until that time comes, however, there is little immediate need to ensure that the simulation is identical to the expected reality.

The simulator is configured by a number of parameters. In particular, the behavior of colliding objects is determined in part by material properties of the objects involved, including restitution (bounciness) and both static and dynamic friction. In our experiments, the same material was used for the robot and the ground plane. Instead of choosing arbitrary material coefficients or trying to determine realistically accurate ones, we repeated our experiments with 27 different configurations; the parameter values are given in Table 1. This provides more data to reinforce our conclusions and, as a side benefit, serves to illuminate the variability one sees when using differently accurate simulators, such as when dealing with the problem of the "reality gap".

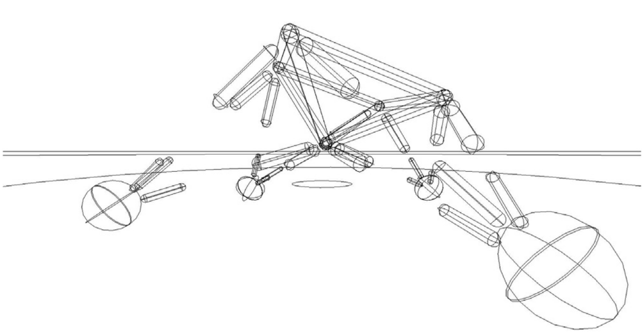

Fig. 2. The PhysX model of the robot. Only the shapes used for collision detection are shown, hence the gaps in the legs where the actuators (which have no shape – they are simply constraints) reside.

Table 1
Fitness comparison. The average fitness of an evolved gait (*EA*) and the fitness of the hand-designed gait (*hand*) is listed for each of the 27 simulator configurations.

Restitution	Static friction	Dynamic friction	EA	Hand
0.0	0.25	0.025	1.79	0.37
		0.125	1.57	0.51
		0.250	1.45	0.64
	1.00	0.100	1.57	0.66
		0.500	1.37	0.70
		1.000	1.32	0.68
	4.00	0.400	1.29	0.50
		2.000	1.27	0.64
		4.000	1.31	0.51
0.4	0.25	0.025	1.78	0.41
		0.125	1.52	0.58
		0.250	1.50	0.65
	1.00	0.100	1.61	0.71
		0.500	1.39	0.73
		1.000	1.32	0.71
	4.00	0.400	1.30	0.72
		2.000	1.31	0.65
		4.000	1.27	0.58
0.8	0.25	0.025	1.76	0.44
		0.125	1.53	0.55
		0.250	1.48	0.64
	1.00	0.100	1.61	0.77
		0.500	1.43	0.84
		1.000	1.38	0.74
	4.00	0.400	1.36	0.80
		2.000	1.36	0.77
		4.000	1.37	0.71

A very simple mechanism was implemented to control the robot's motions. The target length of each actuator was set according to a periodic function of time of the form shown in Fig. 3. Two parameters ∈[0, 1) determine the exact shape of each control signal: the attack phase and the release phase, at which points the target length is set to its maximum and minimum values, respectively. Each actuator tries to drive towards its target length with a constant speed although opposing forces may slow it down. All of the actuator controllers operate at the same fixed frequency of 0.3 Hz but each actuator has its own phase parameters; an entire controller is therefore defined by 24 phase values – two for each of the three actuators in each of the four legs. Such a simple controller was chosen for three reasons: (1) It is observed in nature that undisturbed gaits on flat ground are simply periodic; (2) to make it easier to hand-design gaits for comparison; and (3) to limit the size of the search space for the evolutionary algorithm. Also, simplicity is typically a very good place to start.

To evaluate a candidate controller, it was used to drive the robot for 16 s of simulated time; its fitness was calculated as its average speed over that interval, or zero if the robot fell over. Physics simulation is not particularly cheap – a single evaluation took on the order of 1 s to complete; so, rather than making it an exercise in patience, an effort was made to parallelize evaluations as much as possible. An incremental/steady state evolutionary algorithm (EA) was implemented such that each of the 150 CPUs in a small computing cluster was constantly under heavy load for the duration of

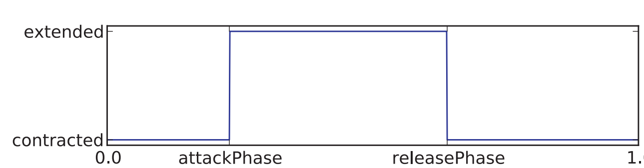

Fig. 3. The actuator control function.