Iterative Product Engineering: Evolutionary Robot Design

Dominic. R. Frutiger[1], Josh C. Bongard[2], Fumiya Iida[2]
[1]Centre of Product Engineering, Swiss Federal Institute of Technology, Zürich, Switzerland
[2]Artificial Intelligence Laboratory, University of Zürich, Switzerland

ABSTRACT

This paper details an iterative, rapid method for digital mock-up and the evolutionary optimisation of a closed loop controller for highly dynamic gaits.
The evolutionary approach in the virtual construction kit MorphEngine is investigated on its capacity to inspire and evolve behaviours for legged robots and non-biometric locomotion. The tool supports the engineer in finding, controlling and finally implementing gaits based on emergent dynamics on a real-world robot through the iterative exploitation of both the agents morphology and its physical environment.

1 INTRODUCTION

A main goal of mobile robotics is the design of an appropriate morphology (sensors, and actuated and passive limbs) together with an adapted control structure that promotes successful locomotion. The principle of natural growth is still unmatched in robotics as it is difficult to design, compute and control complex structures consisting of several interdependent parts (1)(2)(3) and especially if they make use of passive dynamics (4)(5) for rapid locomotion (e.g. freely swinging limbs). Nevertheless, many attempts have been undertaken to address this principle (see chapter 2).
This paper discusses how MorphEngine (see chapter 3) may support the engineer in his investigation of digital mock-ups of legged robots and their gaits based on emergent dynamics by virtue of artificial evolution.
A highly iterative method was used to build a relatively simple monkey-like robot through inspiration drawn from human engineering, evolutionary exploitation of the physical environment and the resulting optimisation of the agent's locomotion pattern in simulation. Eventually this behaviour was transferred to the real world.

2 RELATED APPROACHES

Karl Sims (2) first demonstrated a system in which artificial evolution optimised the morphology and neural controller of a simulated agent to achieve a particular behaviour, including running, jumping, and competing for a common resource.
The Golem Project (3) was the first implementation of evolutionary design of both robot morphology and neural controller coupled to an automated manufacturing process in order to successfully minimize human bias throughout the design process. However, so far, the robot's motors are driven without any sensory feedback. Energy minimization was applied to evolve quasi-static motion in which at each time frame the robot was assumed to be statically stable – thus dynamic locomotion was not investigated.
The design package Darwin2K (1) is another potent tool for real world robotic applications in which human task specification and design constraints are desired and necessary. It was a pioneering approach that included dynamics in the evolutionary optimisation process of an agent's behaviour *and* its morphology.

3 TOOLS

The software package MorphEngine was designed to investigate and verify dynamic locomotion patterns on quickly built digital mock-ups. MorphEngine is a relatively limited tool that does not yet support complex task specifications or an evolutionary optimisation of an agent's structural geometry (see www.ifi.unizh.ch/ailab/people/bongard/MorphEngine/ for documentation and download). Its advantage lies in the user-friendly interface that allows also inexperienced users to construct and investigate arbitrary morphologies with passive or actuated joints including touch-, angle- or light-sensors.
The fitness of the individual depends on its performance on a flat plane (PID and Jacobian-based controllers) and is determined either by the net distance in a specific direction in a given time frame or by its ability to follow a randomly placed light source.
The neural network used to drive the virtual agent is a fully connected, feed forward network. Three input neurons translate the incoming sensory signals into values in [-1.0,1.0]. Values from the input layer, plus a bias neuron, feed into the hidden layer. The hidden layer contains three neurons and is fully, recurrently connected. Values from the hidden layer, plus a bias neuron, feed into two output neurons, which correspond to the two actuated joints. The values of the output neurons are treated as desired joint angles, which are fed into the simulated motors and translated into torque that is applied to the corresponding joint. The values arriving at the hidden and output neurons are summed, and passed through a standard sigmoidal activation function.
Other than in Darwin2K (1) where control parameters are optimised, the evolved controller in MorphEngine consists of a fully connected feed forward neural network with a single layer of recurrently connected hidden nodes. The number of hidden nodes is specified at the beginning of the evolutionary process. Evolution then optimises the synaptic weights of the network in order to achieve successful locomotion (for details see (7)).

4 EXPERIMENTS AND RESULTS

4.1 The Inspiration Factor and Non-Biometric Locomotion

4.1.1 General Investigation of Locomotion Strategies

Various morphologies such as bipeds, quadrupeds, worm-like phenotypes and many more have been built with MorphEngine so far in order to evolve and investigate different locomotion strategies (6)(7). Other current research projects are also inspired by locomotion patterns observed in MorphEngine. One example is the pendulum driven robot Stumpy II (8).

4.1.2 Investigating Tripedal Locomotion

To initially explore the behaviours of morphologies *not* found in nature, a symmetric three-legged creature was built in simulation (fig.1.e: each leg has 3 actuated joint axes equipped with sensors and a touch sensor in the foot), as no biological occurrences of tripedal locomotion are known to the author. As a result, the evolutionary algorithm evolved a control structure for the agent that was well adapted to that specific morphology.

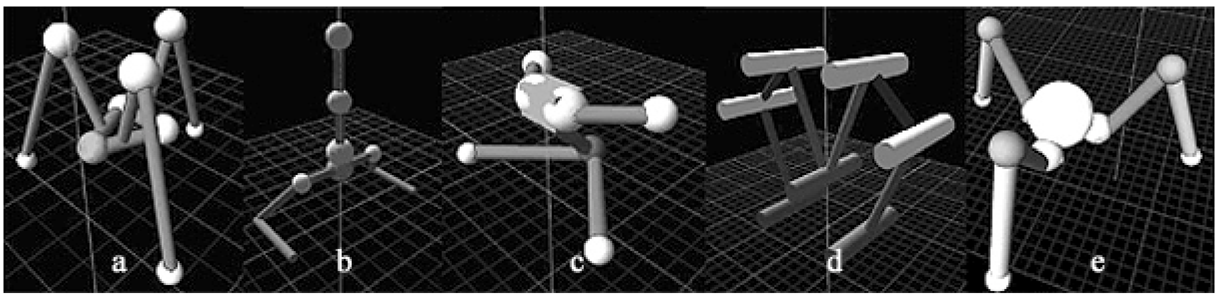

Figure 1. Digital mock-up: investigating locomotion strategies

4.1.3 Results

Many successful and interesting locomotion patterns could be observed (fig.1.a-d: for numerous videos of moving agents please visit www.ifi.unizh.ch/ailab/people/frutiger/ or www.ifi.unizh.ch/ailab/bongard/MorphEngine/OtherVideos/). Simulation made it possible to investigate what kind of morphologies *do* allow successful locomotion (6)(7)(8) that could eventually be implemented in real-world agents (6).

Two different gaits were observed for the tripod-agent (fig.1.e). First, the agent had to follow a randomly placed light source. The resulting phototaxis led to a complex walking pattern that seemed to be regular in one experiment. However, the regularity could not be reproduced in later experiments. Second, the net distance in one specific direction was rewarded: the anticipated result was a jumping behaviour where two legs were used synchronously either to push or pull - depending on the agent's initial orientation - while the third leg acted as the pivot.

4.2 Digital Mock-Up and Prototyping of A Monkey-Type Robot

4.2.1 Implementation and Verification of Closed Loop Locomotion

The transfer of observed, simulated closed loop behaviour to a real robot was tested with the fast and cheap implementation of a simple brachiating agent that takes advantage of the dynamics of a double-pendulum for successful rapid locomotion on a steel rope.

This robot is the first closed loop controlled implementation based on MorphEngine (the creatures of the Golem project (3) and Stumpy II (8) had no sensory feedback from their environment). The goal was to get as close to the simulated behaviour with the real prototype as possible.

4.2.2 Iteratively Investigating and Implementing A Monkey-like Morphology

Step 1: Several classes of monkey-like morphologies were engineered and implemented in MorphEngine to verify that successful locomotion patterns evolved in principle. Those first mock-ups differed mainly in degrees of freedom and claw types and did not yet include any real world constraints such as realistic prototype-scale or motor force (fig.2.a; one example of an early monkey-like series).

Step 2: After confirming that successful locomotion evolved in principle, first hardware considerations were made and implemented with respect to existing motor types and the scale of the future real-world agent. The software had then to be adapted to the new scale of the agents which were much smaller than in preceding purely virtual experiments.

Step 3: Variations of the chosen morphology were engineered with differing numbers and placements of actuated or passive joints and sensors in order to improve the robot's performance (fig.2.b-d: examples of different key-morphology types). Then artificial evolution was applied to select the morphology that showed the most promising locomotion pattern.

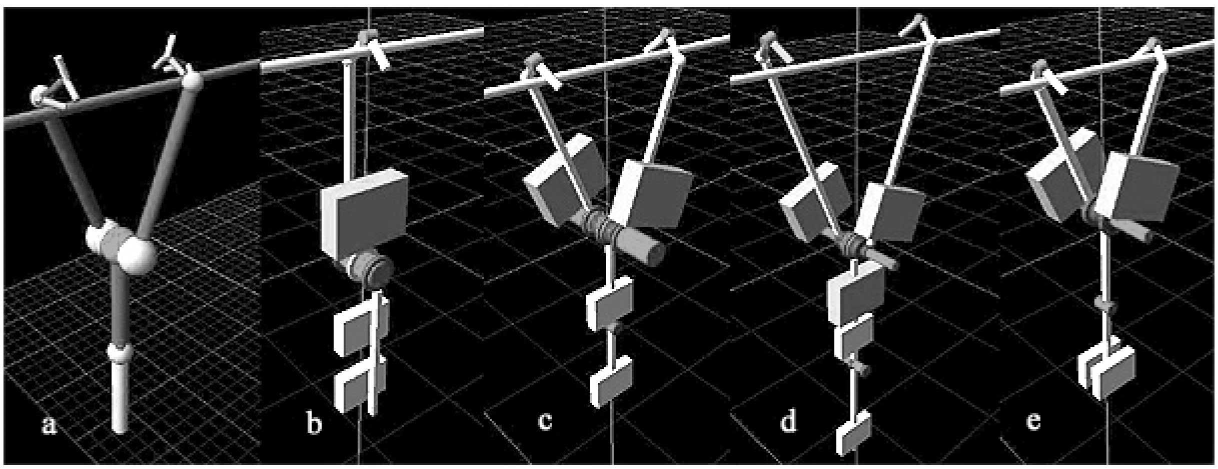

Figure 2. Digital mock-up: investigation of monkey-like morphologies

Step 4: Eventually a morphology was chosen that was simple to construct but interesting enough regarding its passive dynamics to be implemented as a real-world prototype within a short time span of 1-2 weeks (fig.2.c, with two actuated shoulder joints and a passive knee joint: all three include angle sensors).

Step 5: Several iteration steps between hardware considerations such as material density, weight of bearings and other technical details and the translation of these boundary conditions into simulation parameters had to be made. Eventually, the mechanical prototype was built in three days (fig.3); electronics were added in the following week.

Step 6: Finally the real hardware parameters and mass distribution were transferred into simulation again (fig.2.e) in order to fully exploit the new physical properties of both the robot and the environment (fine tuning). The prototype features five sensors in total: a switch in each claw that is activated when an arm is pulled down and pressed on the rope by the weight of the agent and potentiometers that act like angle sensors for each joint.

Step 7: A first successful behaviour could be programmed on the real-world prototype that relied on only three sensors: the angle sensor of the arm initially in the rear (arm A, see fig.4

top) and the touch sensors in the claws. Thus the question arose whether the additional sensors in the second shoulder (arm B, initially in front, see fig.4 top) and in the passive knee joint were really necessary in simulation. An adaptation of the virtual agent to the new settings also led to a successful locomotion pattern in simulation (fig.4 top).

Step 8: The locomotion pattern observed in simulation had to be interpreted and programmed on the prototype, since the direct transfer of a situated artificial neural network is still a technological challenge and was beyond the scope of our approach. The analysis of the motor torque in simulation showed a cycle consisting of roughly seven phases (see *4.2.3 Results, fig.4*).

Step 9: Simple IF- and ELSE-statements on the independent controller boards (one for each motor) controlled the motors (operating at 0.7 Amperes and 8.3 Volts) and were triggered by sensor readings only.

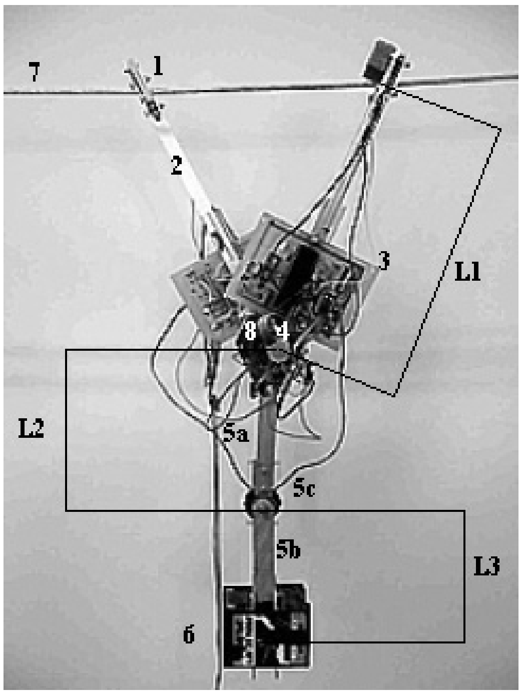

Figure 3. Prototype (side view)

Table 1. Technical Details

Total	**Height ~ 0.54 m / Weight ~ 0.75 kg**
1	Claw [0.015 kg]
2	Arm [L1 = 0.22 m / 0.025 kg]
3	Controller Board [0.054 kg]
4	Motor (ratio 43) [0.068 kg]
5	Body [0.108 kg]
5a	Upper Leg [L2 = 0.15 m]
5b	Lower Leg [L3 = 0.12 m]
5c	Knee Joint (passive)
6	Weight [2 x 0.130 kg]
7	Rope (rubber coated)
8	Shoulder Joint (actuated) [0.026 kg]

4.2.3 Results

The design, simulation and construction of the monkey-like robot was completed in under six weeks. The result was a real-world prototype robot (based on the cross-inspired design) with a successful locomotion pattern similar to the one evolved in simulation that uses the dynamics of a passive double-pendulum (fig.4 bottom).

In *Phase 1* (the phase numbering is visible on the bar in the middle section of fig.4) the virtual agent seems to reduce the friction by pushing its already open arms towards the rope and closing them again with full power in *Phase 2*. This small jump made it possible to pull the arm in the rear forward in *Phase 3* while also accelerating the body. After closing the arms the torque in both shoulders was continuously reduced in *Phase 4* while the momentum gained in *Phase 3* made the body swing forward. The torque changes of the virtual agent in *Phase 5* and *6* were considered to be counterproductive since the robot was not continuously moving but rather 'shivering' at this point (not illustrated). In *Phase 7* the arm in front was successfully moved forward again since all the weight was on the arm in the rear. Eventually, the initial state was reached and the cycle repeated (*Phase 1* and *2*).

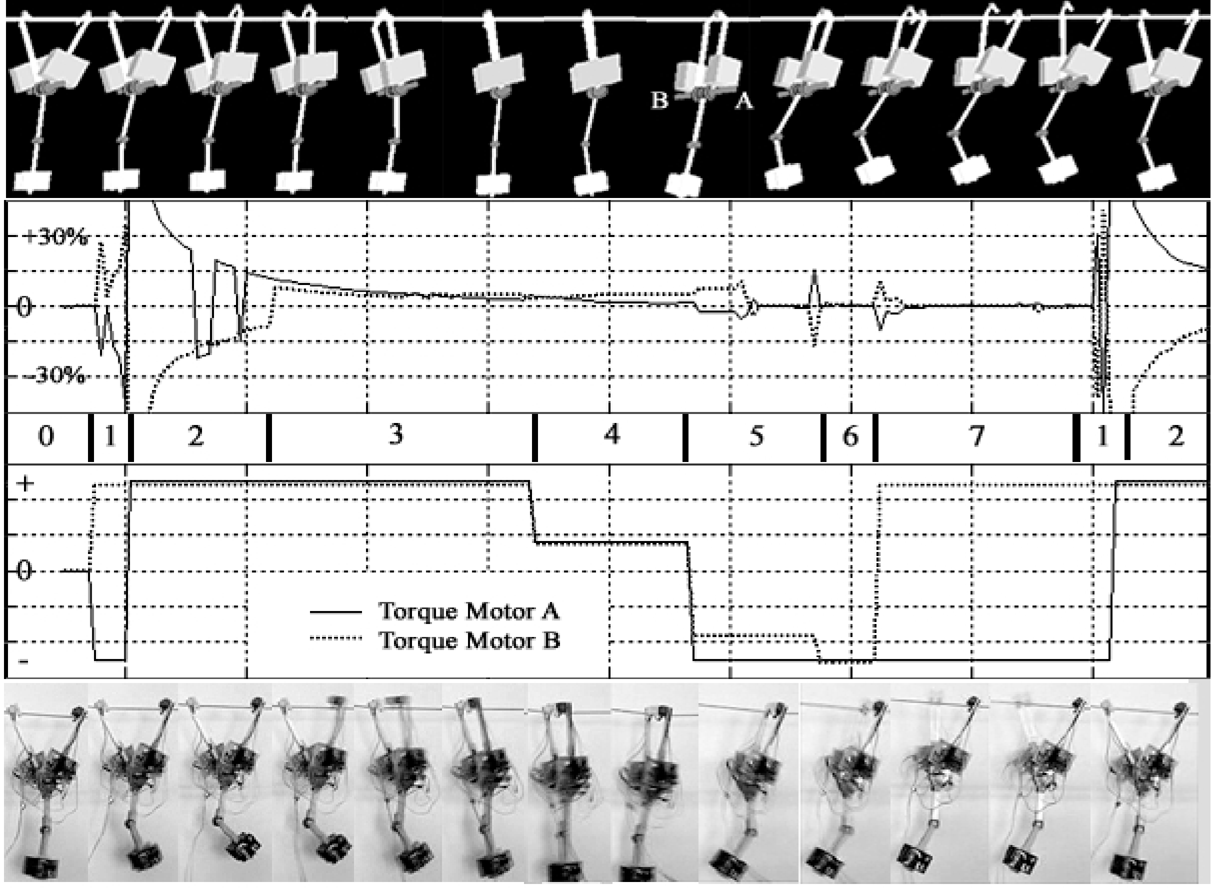

Figure 4. Virtual locomotion cycle (top); qualitative torque analysis and phase definition (centre); real world locomotion cycle (bottom). Agents are moving to the left, the time axes is read to the right. Positive torque-axis oriented towards the reader with the origin in the middle section of the upper body. Motor A pointing towards the reader.

Thus, the creature achieved its movement by changing the centre of gravity to the opposite side of the arm it was about to move by influencing and exploiting the momentum of its pendulum-shaped body. With this strategy, the friction in the moving claw was first reduced and finally totally lost so that the arm could be moved forward freely. This interpretation was implemented in a simpler pattern on the prototype: *Phase 2* had to be skipped since the real agent was simply closing the arms on the spot. In *Phase 5* the centre of mass is actively moved to the rear. Full power is applied in each phase resulting in a faster and much more dynamic movement than in simulation.

5 DISCUSSION

5.1 Non-Biometric Locomotion

When developing walking strategies for the tripedal agent by rewarding for phototactic behaviour, a few successful patterns evolved that seemed regular in two cases although consisting of movements that seemed to be neither very efficient nor particularly elegant from the human point of view. Unfortunately, this minimal regularity could not be reproduced in later experiments and was thus probably just an instance of historical accident during the evolutionary history of that atypical population.

The jumping strategy on the other hand was likely to evolve when rewarding time-efficiency in straight locomotion, since the synchronous movement of two legs implies symmetric locomotion about the direction of travel, and thus the agent moves with a high probability along the shortest path from A to B (energy efficiency was not rewarded). In the case of the tripod a symmetric jumping strategy also led to dynamic stability and was thus preventing the agent from tipping-over on its side (a fate that most agents suffered when trying to follow a light source).

With a few exceptions (such as starfish, a few other radially symmetric animals and crabs), it seems that most organisms are symmetric about a vector corresponding to their usual direction of travel. Considering this, it is not so surprising that the tripedal agents performed well for the forward locomotion task. However, for phototaxis, agents could either turn towards the light source and then move forward again, or they could move towards the target without turning, in which case their line of symmetry no longer matched the direction of travel. It seems that turning was relatively difficult to evolve for these agents, so often the resulting locomotion patterns were irregular, as the agent moved towards the target without aligning one of its three lines of symmetry to that direction.

Symmetry with respect to the main locomotion vector of an agent possibly leads to both energy- and time-efficiency when pure locomotion is the sole behaviour of the robot (3)(9).

5.2 Digital Mock-Up and Prototyping of A Monkey-Type Robot

There are several differences between the locomotion patterns of the virtual and the real agent. Interestingly, one flaw helped to compensate for another: the prototype was brachiating on a rope consisting of twisted steel wires coated with thin rubber while in simulation a perfectly stiff bar was used. The result was a different quality of friction and a dynamic response of the flexible rope. The significantly higher friction between claws and rope in the real world demanded a more dynamic swinging behaviour than observed in simulation (to free the arms). This on the other hand was possible due to the full exploitation of motor torque in reality while the virtual agent was operating at most of the time with only 10-20% percent of its full potential (cut off in fig.4). A possible explanation are known shortcomings of the simulation accuracy mainly in regards to collision detection, material properties (continuous mass distribution, surface structure, friction, stiffness), actuator forces, spring parameters and noise.

5.3 General Aspects

The design with MorphEngine was quick and easy, its capabilities turned out to be sufficient for this project, yet the possibilities of evolution were still quite limited and could have been a bit wider: important features such as evolvable shape and mass of objects or modules, adaptive joint parameters, and realistic material and surface property specification were not implemented. In addition, spheres, cubes and cylinders were the only available objects: the simulation would benefit from the inclusion of non-Euclidean objects. The fitness function was limited to locomotion in one direction or following a light source.

A drawback of the design method was the use of a programmed computer chip instead of a directly implemented artificial neural network. Having no possibilities for a direct transfer yet, the question arises, how well a resulting locomotion pattern in general can be understood and interpreted by a human engineer. As the brachiating robot constitutes a rather simple example of dynamic locomotion it is not clear whether a similar approach will work for more complex patterns when the interactions are no longer obvious and parallel processing would be required for real time performance. One of the main weaknesses of such an iterative approach is the unpredictability of the impact of parameter changes such as population size, number of generations, mutation rate and number of hidden nodes in the neural network on the evolutionary algorithm.

6 CONCLUSIONS

The results presented are promising, rapid dynamic locomotion was successfully implemented on a real-world robot. Yet there is much to be done in order to improve the accuracy and capabilities of the design tool MorphEngine for rapid, low-tech robot implementations and quick locomotion pattern verifications. One may state that the solutions the algorithm came up with were obvious and that the exploitation of physical properties of the robot and the environment was to be expected. This might be true in many cases and often the evolving behaviour was in fact anticipated. However, such a tool may be of a great benefit to an engineer or roboticist as it comes to designing and rapidly evaluating new solutions for a complex structure with many moving parts and complex mass distributions.
With respect to a professional engineering approach aimed at a real-world industrial product, Darwin2K (1) might be the better choice due to the still very limited capabilities of MorphEngine (see paragraph 4.3). Unfortunately, there were no real-world implementations designed with Darwin2K up to the appearance of this paper to underline the potential of that tool in this respect.

REFERENCES

1 **Leger, C.**, (2000), Darwin2K: An Evolutionary Approach to Automated Design for Robotics. Kluwer Academic Publishers, New York.

2 **Sims, K.**, (1994) "Evolving 3D morphology and behaviour by competition", in *Artificial Life IV*, pp. 28-39.

3 **Lipson, H.** and **Pollack, J. B.**, (2000), "Automatic design and Manufacture of Robotic Lifeforms", *Nature* 406, pp. 974-978

4 **Yim, M., Zhang, Y., Lamping, J., Mao, E.**, (2001) "Distributed Control for 3D Metamorphosis", Autonomous Robots, 10:1, pp. 41-56.

5 **Pratt, J., Pratt, G.**, (1999), "Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation.", in *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99)*, Portsmouth, UK.

6 **Paul, C., Bongard, J. C.**, (2001), “The Road Less Travelled: Morphology in the Optimization of Biped Robot Locomotion”, in *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, Hawaii, USA.

7 **Bongard, J. C., Pfeifer, R.**, (2002), “A Method for Isolating Morphological Effects on Evolved Behaviour”, in *Proceedings of the Sixth European Conference on Artificial Life*, Prague, CZ, pp. 401-412.

8 **Paul, C., Dravid, R.** and **F. Iida**, (2002), "Design and Control of a Pendulum Driven Hopping Robot ", to appear in *The IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland

9 **Bongard, J. C., Paul, C.**, (2000) "Investigating Morphological Symmetry and Locomotive Efficiency using Virtual Embodied Evolution", in J.-A. Meyer et al (eds.), *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour*, pp. 420-429.