Locomotion of Modular Robots:

Optimizing Modular Robots Locomotion in Simulation and Applying Results to Real-World Robot

BIOROB
EPFL Biorobotics Laboratory

Yura Perov[1] (yura.perov@epfl.ch), **Rico Möckel**[2], **Massimo Vespignani**[2], **Soha Pouya**[2], **Stéphane Bonardi**[2], **Auke Jan Ijspeert**[2]
[1]Siberian Federal University, Institute of Mathematics, [2]Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Research Object: Roombots

Developed at the BioRob lab, **Roombots** are designed to be

- **modular** — they are composed of uniform modules,
- **self-reconfiguring** — modules can attach to and detach from each other in order to form different shapes of the whole robot.

These kinds of robots is potentially more **robust** and **adaptive** than monolithic robots, because they can adjust to different tasks and environments by changing their shapes. Because modules are homogenous and modular, these robots become **cheaper** (through scale production), and their broken parts can be easily replaced.

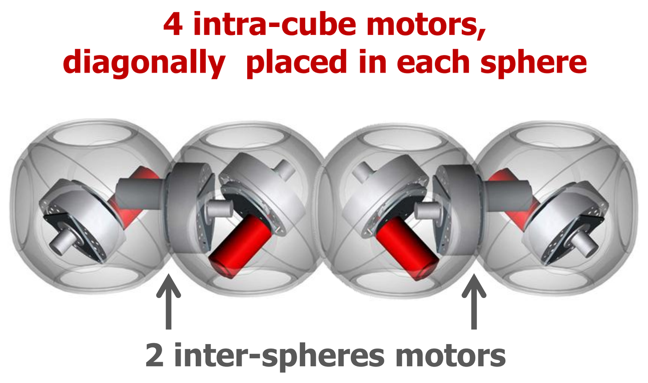


Figure 1. In our experiments the Roombots **metamodule (4 spheres)** was used [5].

Figure 2. The corresponding (to 6 motors) **Central Pattern Generator (CPG) neural network**. CPG is a simple, but flexible parameterized movement model. It was discovered in the neuroethology that CPG serves many functions in vertebrate animals.

Motivation

How to find proper angular commands (that will be sent to the Roombots' motors) to generate **fast robot locomotion (gait)**? How to find a general solution to teach the robot to move as rapidly as possible:

- for **all** possible robot's architectures,
- in significantly **different** environments?

Rejected approaches

Engineers can manually design fast gaits, but it is **expensive**, **not generic** and, probably, will **not work** for **another** environment and/or robot architecture.

Inspired by nature we can use **stochastic optimization**. Firstly, we determine the general pattern of **parameterized movement model**, and, secondly, wisely search for a good set of movement model parameters.

However, the resultant stochastic optimization usually requires too many calculations of the fitness function, i.e. **excessively many testings** of different movement model parameters on **the hardware robot**. It is time consuming and expensive, as it wears down the hardware robots very rapidly.

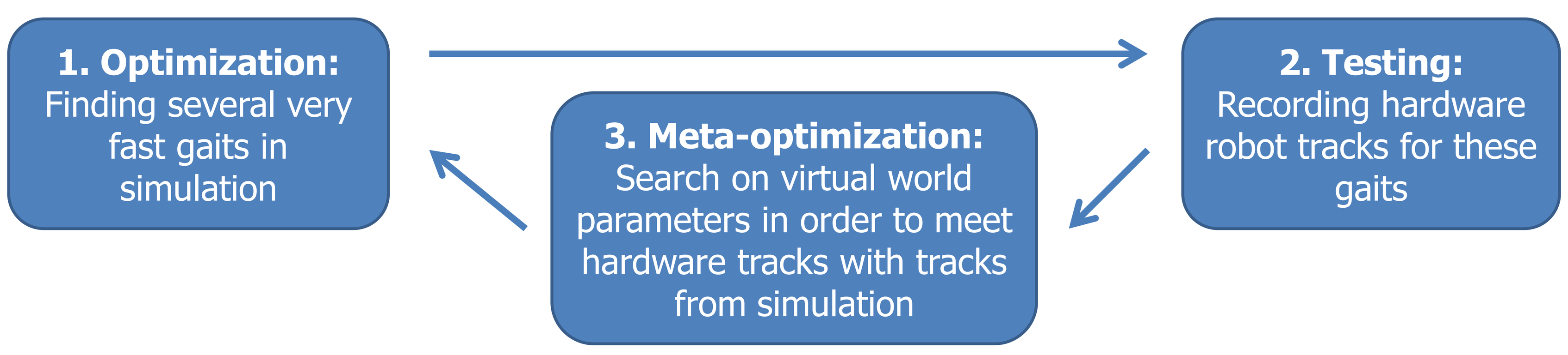
But why not doing this optimization with the help of **virtual simulation**? (Inspired by [2] and its references.)

Optimization in Simulation

First hypothesis: The virtual model of the robot and the simulated world should correspond to the real robot and the real world environment, but slight simplifications/inaccuracies are allowed. The noise addition to the physical simulation engine helps to obtain robust gaits [3].

Second hypothesis: If it is hard to specify environmental properties (coulomb friction, elasticity coefficients, etc.), the circles of optimization and meta-optimization could be run [1].

Circles of optimizations



Results and Contribution

1. **Significant improvements** are made to the virtual robot model, i.e. the model became **more consistent** with the real hardware model.

Figure 3. The virtual model of the robot and the real hardware robot.

2. Suggested several algorithms for the optimization on CPG movement model parameters for search of fast gaits.

$$FitnessFunction = \frac{\sum_{test=1}^{3} Speed}{3} - Max(Errors) \rightarrow max$$

$$Errors = \{Distance(Position_{test\_i;time}, Position_{test\_j;time}) | time = \overline{1,30}, test\_i, test\_j = \overline{1,3}\}$$

Figure 4. Formulae of the best optimization's fitness function.

3. One optimization algorithm showed significant result. With its help a **fast impactless Roombots gait was found** (speed 8 cm/sec) in simulation, and then without any tuning this gait worked fine on the hardware robot (6 cm/sec). This gait is currently the fastest gait for Roombots metamodule, including all previously found gaits, mostly during hardware optimization [5].
4. Improved the hardware robot tracking system, making it possible to save the movement trajectory of the geometrical center of the real robot image. Recorded many different real robot tracks in order to do the meta-optimization.

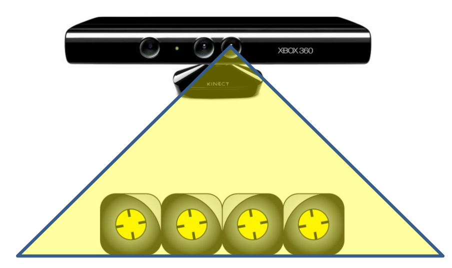


Figure 5. The hardware robot is tracked from above with the Microsoft Kinect depth camera.

5. Carefully selected 20 parameters for the meta-optimization.
6. Suggested first very simple algorithm for the meta-optimization, it has been tested and failed. It shows that it is dangerous to oversimplify.
7. Suggested and implemented a new robot tracking approach to compare real and simulated robot tracks. It is based on the **overlapping comparison** of captured from above hardware and simulation robot positions with simplified Newton inter-optimization.

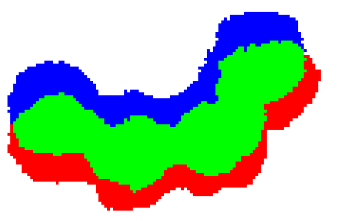


Figure 6. Overlapped Roombots "from above". Blue points correspond to the virtual Roombots from simulation, red points to the real hardware Roombots, green points are points of the overlapping region. The rate of overlapping in percentage terms (the more, the better) is the ratio of green points to all variegated points.

8. Real robot tracks for the new meta-optimization approach were recorded. Latest results show a significant improvement of **45 %** in the similarity of real world and virtual world. The *training-test sets approach* confirmed the result, rejecting the hypothesis of the *over*-meta-optimization. In addition, the speed of the above-mentioned best found gait, which has speed **5.8 cm/sec** on the hardware Roombot, decreased to **6.0 cm/sec** in the meta-optimized virtual world versus *8.0 cm/sec* in the initial virtual world without meta-optimization.

Conclusion

1. We showed for the modular robots, Roombots, the **possibility of optimizing** the CPG movement model parameters in **pure simulation** with noise addition, and **successfully applying** received fast gaits on the Roombots **hardware** without any intermediate forced tuning.
2. We also showed that in case of necessity of the virtual world parameters adjustment (if your initial model is too rough for the optimization on gaits speed), **optimization—meta-optimization circles** could be successfully used in order to match *enough* the virtual model to the real world.

References

1. H. Lipson, J. C. Bongard, V. Zykov, and E. Malone, "Evolutionary robotics for legged machines: From simulation to physical reality," in Proceedings of the 9th Int. Conference on Intelligent Autonomous Systems, pp. 11–18, 2006.
2. K. Glette, G. Klaus, J. C. Zagal, and J. Torresen, "Evolution of locomotion in a simulated quadruped robot and transferral to reality," in Proceedings of the 17th Int. Symposium on Artificial Life and Robotics, 2012.
3. N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life, pp. 704–720, Springer-Verlag, 1995.
4. S. Coros, A. Karpathy, B. Jones, L. Reveret, and M. van de Panne, "Locomotion skills for simulated quadrupeds," in ACM SIGGRAPH 2011 papers, ACM, pp. 59:1–59:12, 2011.
5. T. A. Nguyen, "Online Optimization for the locomotion of Roombots", semester project at BioRob lab, EPFL, 2012 (figures 1, 2, 5 were taken from this reference).
6. F. Wilhelm, "Online optimization for the locomotion of Roombots structures," semester project at BioRob lab, EPFL, 2012.
7. Official Roombots web-page, "http://biorob.epfl.ch/roombots", BioRob lab, EPFL, 2012.