Lessons Learned from Real-World Experiments with DyRET: the Dynamic Robot for Embodied Testing

Tønnes F. Nygaard, Jørgen Nordmoen, Charles P. Martin and Kyrre Glette
Department of Informatics and RITMO, University of Oslo, Norway
Email: tonnesfn@ifi.uio.no

I. INTRODUCTION

Robots are used in more and more complex environments, and are expected to be able to adapt to changes and unknown situations. The easiest and quickest way to adapt is to change the control system of the robot, but for increasingly complex environments one should also change the body of the robot – its morphology – to better fit the task at hand [1].

The theory of Embodied Cognition states that control is not the only source of cognition, and the body, environment, interaction between these and the mind all contribute as cognitive resources [2]. Taking advantage of these concepts could lead to improved adaptivity, robustness, and versatility [3], however, executing these concepts on real-world robots puts additional requirements on the hardware and has several challenges when compared to learning just control [4].

In contrast to the majority of work in Evolutionary Robotics, Eiben argues for real-world experiments in his “Grand Challenges for Evolutionary Robotics” [5]. This requires robust hardware platforms that are capable of repeated experiments which should at the same time be flexible when unforeseen demands arise.

In this paper, we introduce our unique robot platform with self-adaptive morphology. We discuss the challenges we have faced when designing it, and the lessons learned from real-world testing and learning.

II. THE ‘DyRET’ ROBOT

Our robot, DyRET (Dynamic Robot for Embodied Testing), was developed to be a platform for experiments on self-adaptive morphologies and embodied cognition [6], shown in Fig. 1. It is a fully certified open source hardware project, and documentation, code and design files are freely available online [7]. Since it is intended for use with machine learning techniques it is designed to be robust, withstanding falls from unstable gaits [8]. It can actively reconfigure its morphology by changing the lengths of its femur and tibia, which can be used to mechanically gear the motors, and allow the robot to change the trade-off between movement speed and force surplus continuously [9].

III. EXPERIENCES AND CHALLENGES

In this section, we present some key lessons we have learned when working with DyRET. We have tried to summarize the lessons, followed by more detailed explanations.

Robustness and maintainability are more important than ease of building. Using rapid prototyping and design for manufacturability principles, along with exploiting COTS components is crucial in achieving an effective design process of a legged robot.

Legged robots are very complex systems, and anticipating all demands and challenges early in the design process is impossible. Techniques from rapid prototyping allowed us to quickly get physical prototypes of the robot, which allowed us to see and fix challenges that would be difficult to find without having physical proof-of-concept models of the system available. An important part of this, is to use already existing Commercial-off-the-shelf (COTS) components where available. This allows us to capitalize on the work of others, and also makes it easier for others to build or utilize lessons learned from our designs. Design for manufacturability is another important concept, and promotes adapting the design to manufacturing considerations during the initial design process, where they can be solved much more easily than during operation. Making a robot that is easy and cheap to build can be important, but our experience is that maintainability is even more important, especially when using machine learning that puts huge strains on the physical robot.
Repairs and mechanical failures

A good strategy for redesign is to balance quick spot repairs and laborious systematic analyses of failures. Increasing the strength of individual parts that break is often not an effective way to do iterative design.

Designing parts for legged robots is always a trade-off between strength and weight, and mechanical failures during prototyping is guaranteed. Strengthening the part that broke can be a quick fix, but our experience is that this often moves the problem. Both high persistent forces and sudden shock travel through the mechanical design, and lead to failure in the next weakest link of the chain. Reducing stress concentrations locally in a particular part can sometimes be successful in allowing the robot to withstand a similar situation again, however, excessive force can often lead to cascading failures throughout the system. Having a clear strategy for when and what to do when mechanical failures happen is important, and early on deciding on a balance between quick spot repairs and laborious systematic analyses of failures. Once an experiment is underway, replacing parts with similar parts might be the only option without skewing the results, so extra efforts on failure identification during the prototyping phase might be worth the effort. Larger cracks in the material are often easy to identify, but deflection during operation, small fractures, or material creep can be harder to detect.

Controller complexity

Low controller complexity often entails ease of understanding with less dynamic results, while high complexity controllers can give good results, but demand more optimization. In hardware this trade-off is worth thinking about.

Learning legged locomotion, or a gait, is a difficult challenge. To optimize the gait, the movement of the legs is parameterized through a gait controller. A lot of a priori knowledge can be embedded into the controller, and result in few parameters that are easy to optimize. Less prior knowledge requires more of the optimization algorithm, resulting in an increased number of evaluations. The more knowledge that is embedded, the less room there is for a varied range of behaviors, which might be needed to adapt to new or changing tasks, environments or the robot itself [10]. Finding the right complexity balance can be very challenging, especially in real-world learning where the number of evaluations are limited. We have successfully used a gait controller with dynamic complexity [11], but similar performance can be achieved by focusing on complexity early on in the process. Another option is using different controllers for different environments or tasks [12], for instance a complex controller when optimizing the gait in a simulator with cheap evaluations, and a less complex controller in the real world.

Experiment design

Both the environment and the robot itself are dynamic, and changes will happen during operation. This can lead to biases in the experiment results, which have to be controlled for by proper experiment design.

One of the key insights we have experienced after several papers worth of real-world experiments on DyRET is how components change characteristics during the course of experiments. Because of this gradual change it is important to store as much information as possible so that automatic procedures can be applied to detect differences during and after experiments. A big difference between simulation and real-world experiments is that a real-world experiment can never be perfectly replicated. The change in characteristics should also guide the experiment design in the real world. Because components are expected to change it is important to evenly test different solution so as to not bias the experiment towards a specific solution. A concrete example is the reduction in performance of our joints as the motors heat up. If the solutions are always tested in the same order, this might affect the results, and give spurious effects that cloud the results.

Starting in the real world

Starting with simulation can be a quick way to get started learning locomotion, however, it is more difficult to transition from abstract simulated robots to the real world compared to going from a physical system to simulation.

Evaluating solutions on a physical robot system can take several seconds to minutes, depending on gait complexity and experiment design. Evaluating on physics simulations or with simplified models, often done in software, can give a speedup of several orders of magnitude. This often makes simulation a flexible and easier starting point. However, our experience with DyRET indicates that going from a real-world robot to simulation can yield more realistic simulation results which in turn translates to more sensible real-world gaits after software optimization. Not basing a virtual robot on a physical prototype makes it easier to make choices resulting in solutions that turn out to be infeasible in the real world. [13].

IV. CONCLUSION

In this short abstract we have presented lessons learned through several years of experimentation on the DyRET platform, which have resulted in 13 scientific publications. From initial design considerations to the challenges, such as the trade-off between simulated experiments and real world learning, which have guided the choices we have taken. We hope to encourage more researchers within the robotics community to try real-world experiments and by sharing knowledge usually not found in publications we believe getting started with hardware could be easier.
REFERENCES

[1] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, “Exploring mechanically self-reconfiguring robots for autonomous design,” 2018 ICRA Workshop on Autonomous Robot Design, 2018. [Online]. Available: [http://arxiv.org/abs/1805.02965](http://arxiv.org/abs/1805.02965).

[2] A. Wilson and S. Golonka, “Embodied cognition is not what you think it is,” Frontiers in Psychology, vol. 4, p. 58, 2013.

[3] J. Nordmoen, T. F. Nygaard, K. O. Ellefsen, and K. Glette, “Evolved embodied phase coordination enables robust quadruped robot locomotion,” in Proceedings of the Genetic and Evolutionary Computation Conference. ACM, 2019.

[4] T. F. Nygaard, E. Samuelsen, and K. Glette, “Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach,” in Applications of Evolutionary Computation. Springer, 2017, pp. 825–836.

[5] A. E. Eiben, “Grand challenges for evolutionary robotics,” Frontiers in Robotics and AI, vol. 1, p. 4, 2014.

[6] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, “Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing,” in 2019 IEEE International Conference on Robotics and Automation (ICRA), 2019.

[7] T. F. Nygaard and J. Nordmoen, “DyRET software repository,” [https://github.com/dyret-robot/dyret](https://github.com/dyret-robot/dyret) documentation, 2019.

[8] T. F. Nygaard, J. Torresen, and K. Glette, “Multi-objective evolution of fast and stable gaits on a physical quadruped robotic platform,” in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Dec 2016, pp. 1–8.

[9] T. F. Nygaard, C. P. Martin, E. Samuelsen, J. Torresen, and K. Glette, “Real-world evolution adapts robot morphology and control to hardware limitations,” in Proceedings of the Genetic and Evolutionary Computation Conference. ACM, 2018.

[10] J. Nordmoen, K. O. Ellefsen, and K. Glette, “Combining map-elites and incremental evolution to generate gaits for a mammalian quadruped robot,” in Applications of Evolutionary Computation, K. Sim and P. Kaufmann, Eds. Springer International Publishing, 2018, pp. 719–733.

[11] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, “Evolving robots on easy mode: Towards a variable complexity controller for quadrupeds,” in Applications of Evolutionary Computation. Springer, 2019.

[12] J. Nordmoen, E. Samuelsen, K. O. Ellefsen, and K. Glette, “Dynamic mutation in map-elites for robotic repertoire generation,” in Artificial Life Conference Proceedings. MIT Press, 2018, pp. 598–605.

[13] J.-B. Mouret and K. Chatzilygeroudis, “20 years of reality gap: a few thoughts about simulators in evolutionary robotics,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion. ACM, 2017, pp. 1121–1124.