Experiences from Real-World Evolution with DyRET: Dynamic Robot for Embodied Testing*

Tønnes F. Nygaard[1], Jørgen Nordmoen[1], Kai Olav Ellefsen[1], Charles P. Martin[1,2], Jim Tørresen[1,2], and Kyrre Glette[1,2]

[1] Department of Informatics
University of Oslo, Norway
tonnesfn@ifi.uio.no
[2] RITMO Center of Excellence
University of Oslo, Norway

Abstract. Creating robust robot platforms that function in the real world is a difficult task. Adding the requirement that the platform should be capable of learning, from nothing, ways to generate its own movement makes the task even harder. Evolutionary Robotics is a promising field that combines the creativity of evolutionary optimization with the real-world focus of robotics to bring about unexpected control mechanisms in addition to whole new robot designs. Constructing a platform that is capable of these feats is difficult, and it is important to share experiences and lessons learned so that designers of future robot platforms can benefit. In this paper, we introduce our robotics platform and detail our experiences with real-world evolution. We present thoughts on initial design considerations and key insights we have learned from extensive experimentation. We hope to inspire new platform development and hopefully reduce the threshold of doing real-world legged robot evolution.

Keywords: evolutionary robotics · real-world evolution · lessons learned

1 Introduction

Robots are used in more and more complex environments, and are expected to be able to adapt themselves 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]. To achieve this vision, researchers need access to flexible robot platforms that can be adapted to

* This work is partially supported by The Research Council of Norway under grant agreement 240862 and through its Centers of Excellence scheme, project number 262762. Simulations with DyRET were performed on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway.

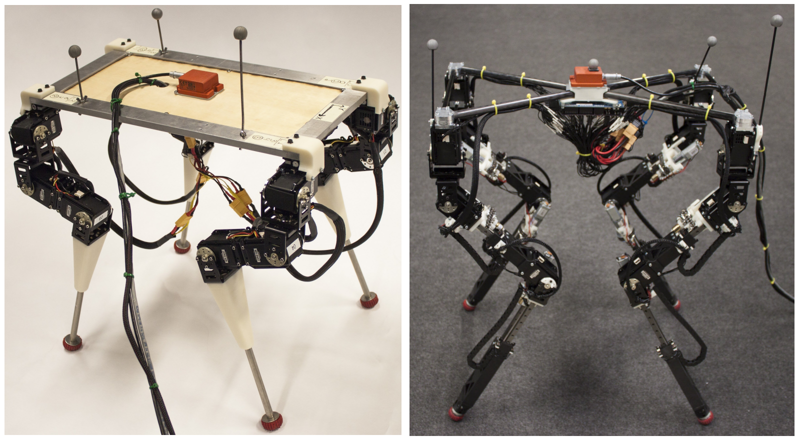

Fig. 1: Initial version of DyRET (left) without self-modifying legs. Latest version of DyRET (right) with fully extended legs.

new environments and tasks. For many projects this limits choices to simulated experiments on virtual robots that are never tested in the real world.

Evolutionary Robotics takes inspiration from natural evolution, with concepts such as hereditary traits and genome mutation, and applies these principles to robotics. This combination has shown incredible creativity, not only creating novel robot controllers but, going as far as creating whole new robot bodies. However, this creativity is usually constrained to the software realm due to the ease of simulating these new creations and the difficulty in performing the same number of experiments in the real world. In contrast to the majority of work in Evolutionary Robotics, Eiben argues for real-world experiments in his "Grand Challenges for Evolutionary Robotics" [2]. This requires robust hardware platforms that are capable of repeated experiments. At the same time, these platforms must be flexible to manage unforeseen demands.

An emerging concept within evolutionary robotics is the theory of *Embodied Cognition*. This theory suggests that reasoning and cognition cannot be fully understood if studied in simple computer models alone. The mind, body, environment, and the interaction between these all contribute as cognitive resources [3]. Taking advantage of these concepts could lead to improved adaptivity, robustness, and versatility [4], however, executing these concepts on real-world robots puts additional requirements on the hardware and raises several challenges when compared to learning just control [5].

In this paper, we will present related work before introducing our robot platform with self-adaptive morphology, seen in Fig. 1. The main section of the paper will describe the challenges we have faced when designing the robot, and the lessons learned from real-world evolution and experimentation. By summarizing our experiences we can report on key insights which can hopefully lead to better robotics platforms in the future.

2 Background

Robots are becoming a more widely used tool in many industries, and are used for advanced tasks and in complex environments. Historically wheeled robots have been used extensively in industrial settings because of their simplicity and ease of deployment [6]. However, we are now starting to see the need for robots to operate in more complex environments, both inside and out in the real world [7]. Using legs instead of wheels allows the robot to traverse difficult terrains and environments, making the robot accommodate the user instead of requiring the user to adapt to the robot.

2.1 Evolutionary robotics

The field of Evolutionary Robotics (ER) uses techniques from evolutionary computation to optimize both a robot's control and body [8]. Many different legged robots have been used in ER research. Some use off-the-shelf standard robots not specifically designed for ER research, like Sony's Aibo [9], while others use robots specifically built for the purpose, like the Aracna [10].

Most earlier work in ER only optimize the control system of the robot [2]. This can allow the robot to adapt to the environment it is operating in [11], or to changes to the robot itself [12]. However, only changing the control has its limitations, and earlier work has shown that changing the morphology yields results that could not be achieved by changing control alone [13]. Furthermore, most work is done on virtual robots in simplified physics simulations, and not on actual physical robots [14]. This allows for simple parallelization and noise-free evaluations, but the inaccuracies in the simulator or models used often lead to big discrepancies in the performance of the virtual robot and its real world counterpart [15]. There are many techniques to reduce this reality gap [16], but even with recent strides, this is becoming more and more challenging, as both the robots themselves, the environment they operate in, and the tasks they are solving become more complex and harder to model.

2.2 Embodied cognition

The theory of Embodied Cognition originally came from psychology, but is making its way into many sub-fields of robotics, including swarm robotics and modular robotics [17]. The original theory states that the brain is not the only cognitive resource a human has, and that the body, the environment, and the interactions between these can also serve as cognitive resources[3]. There are several examples where this has been used successfully in robotics [18]. An important aspect of this approach, is that a large part of the cognition, or problem solving ability of a robot, can be placed in the robot body, its environment, and the interactions these form with each other and the robot controller. Therefore, inaccurate models of either environment or body can make it impossible to do this on anything but the physical robot in the real world [2].

3 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 [19], shown in Fig. 2. It is a fully certified open source hardware project, and documentation, code and design files are freely available online [20]. Since it is intended for use with machine learning techniques it is designed to be robust, so that is can withstand falls from unstable gaits [21]. It can actively reconfigure its morphology by changing the lengths of its femurs and tibias. Shorter leg length increases the force at the end of the leg, given constant torque from the servo. The self-changing morphology therefore allows the robot to change the trade-off between movement speed and force surplus continuously, and can serve as a gearing of the motor [22].

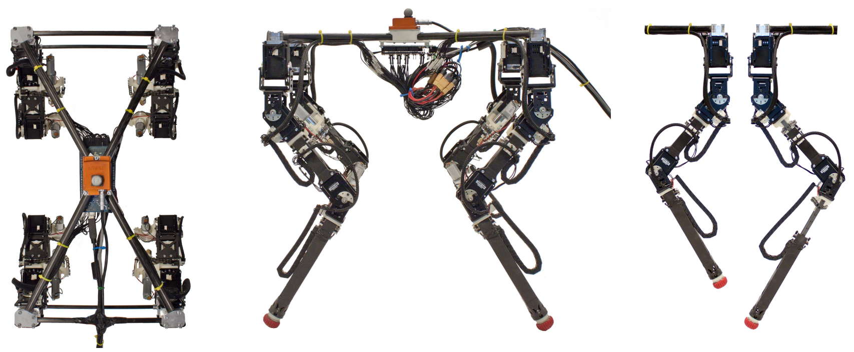

Fig. 2: Top and left views of our reconfigurable robotic platform, and examples of the legs at two different lengths.

The robot is built using *Commercial-off-the-shelf* (COTS) components where possible, and all custom parts can be made with consumer grade 3D printers. We also use composite tubing for structural integrity. Selected parts have an alternative design in aluminium for more demanding requirements, and have been milled. Dynamixel servos are used in all rotational joints, which feature on-board PID controllers for accurate position control. The servos are connected to a common bus that interfaces to a computer over USB. The length of each leg segment is controlled by a custom linear actuator, driven by a standard DC motor. The main mechanism consists of a lead screw that moves carriages along two rails using a chain, all COTS components. An encoder gives the position of each actuator, and a simple positional controller is run on an Arduino Mega board with a custom interface shield. The robot features an XSense MTi-30 Attitude and Heading Reference System that measures linear acceleration, rotational velocity and absolute orientation. The robot has reflective markers that is used with motion capture equipment to get the absolute position of the robot. It also features directional force sensors mounted on each foot which can be used to detect when the feet touch the ground.

4 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.

Initial design considerations

Robustness and maintainability are more important than ease of building. Using *rapid prototyping* and *design for manufacturability* principles, along with exploiting *Commercial-Off-The-Shelf* components are 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. As an example of this we have included the designs in Fig. 3 which illustrates how the manufacturing methods should help inform the design of the individual parts. 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 considerable strains on the physical robot.

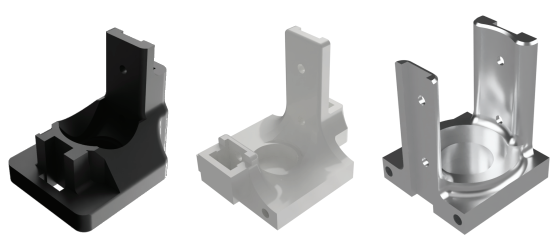

Fig. 3: The two parts on the left are designed for two different 3D printers while the part on the right is designed for milling. This is an example of designing for manufacturability where parts are designed for the same purpose, but optimized for different manufacturing methods.

Repairs and mechanical failures

A good strategy for redesign is important 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 results in the problem being transferred to other parts of the robot. Both high persistent forces and sudden shock travel through the mechanical design, and lead to failures 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. An example of this can be seen in Fig. 4, where a strengthening of a part that broke lead to the next part in the chain breaking instead. 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.

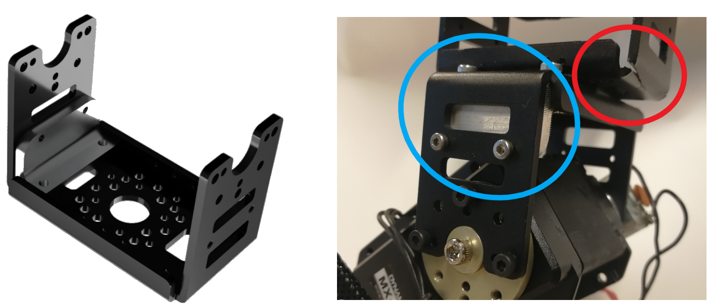

Fig. 4: An example showing a cascading mechanical failure, where an initial strengthening of a broken part (left, and circled blue on the right) leads to a failure in the next part in the chain (red circle).

Controller complexity

Low controller complexity puts less strain on the robot by testing solutions that are safer and more conservative, and is quicker to optimize. High complexity controllers can give better results by having higher freedom, but will necessarily test solutions that are incautious. This complexity trade-off is often not considered when doing simulation-only experiments, but can be imperative when working on physical robots.

Learning legged locomotion is a difficult challenge. To optimize the walking pattern, the gait, the movement of the legs is parameterized through a gait controller. Much a priori knowledge can be embedded into the controller, resulting 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 [23]. 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 [24], which can be seen in Fig. 5. Another option is using different controllers for different environments or tasks [25], for instance a complex controller when optimizing the gait in a simulator with cheap evaluations, and a less complex controller in the real world.

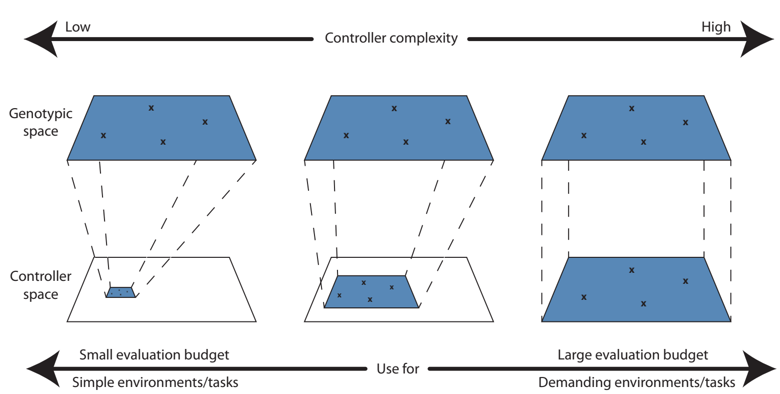

Fig. 5: Diagram of a controller with adaptable gait complexity. Here, a dynamic genotype-phenotype mapping allows a single parameter to control the complexity of generated gaits [24].

Starting in the real world

Using a virtual robot can be a quick way to get started learning locomotion. It is, however, 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 seconds to several minutes, depending on gait complexity and experiment design. Evaluating in 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 [14], illustrated in Fig. 6.

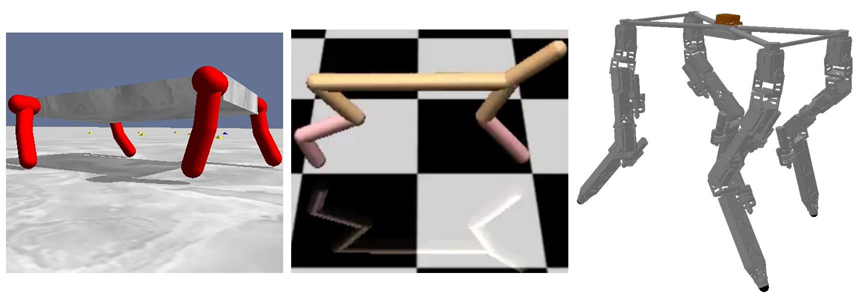

Fig. 6: Comparison of different legged robots in simulation, with DyRET on the right. Since DyRET was first designed in hardware and then transferred to simulation it is known that it would function in the real world after simulation.

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 by proper experiment design.

One of the key insights we have experienced after 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 solutions so as to not bias the experiment towards a specific one. A specific 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 can give noise in the results.

5 Conclusion

In this paper we have presented lessons learned through extensive experimentation on the DyRET platform. This includes both initial design considerations, and challenges such as the trade-off between simulated experiments and real world evolution. Having a mechanically self-modifying quadruped robot is rare among platforms used in evolutionary robotics research. This gives us a unique insight into evolution of control and morphology in the real world. By sharing knowledge usually not found in experiment-based publications, we hope to encourage more researchers within the evolutionary robotics community to try real-world experiments.

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.
2. A. E. Eiben, "Grand challenges for evolutionary robotics," *Frontiers in Robotics and AI*, vol. 1, p. 4, 2014.
3. A. Wilson and S. Golonka, "Embodied cognition is not what you think it is," *Frontiers in Psychology*, vol. 4, p. 58, 2013.
4. 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.
5. 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.
6. M. F. Silva and J. Tenreiro Machado, "A historical perspective of legged robots," *Journal of Vibration and Control*, vol. 13, no. 9-10, pp. 1447–1486, 2007.
7. J. E. Bares and W. L. Whittaker, "Configuration of autonomous walkers for extreme terrain," *The International Journal of Robotics Research*, vol. 12, no. 6, pp. 535–559, 1993.
8. S. Doncieux, N. Bredeche, J.-B. Mouret, and A. E. G. Eiben, "Evolutionary robotics: what, why, and where to," *Frontiers in Robotics and AI*, vol. 2, p. 4, 2015.
9. G. S. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita, "Evolving robust gaits with aibo," in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2000, pp. 3040–3045.

10. S. Lohmann, J. Yosinski, E. Gold, J. Clune, J. Blum, and H. Lipson, "Aracna: An open-source quadruped platform for evolutionary robotics," in *Artificial Life Conference Proceedings 12*. MIT Press, 2012, pp. 387–392.
11. E. Haasdijk, N. Bredeche, and A. E. Eiben, "Combining environment-driven adaptation and task-driven optimisation in evolutionary robotics," *PLOS ONE*, vol. 9, no. 6, pp. 1–14, 06 2014.
12. S. Koos, A. Cully, and J.-B. Mouret, "Fast damage recovery in robotics with the t-resilience algorithm," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1700–1723, 2013.
13. G. Picardi, H. Hauser, C. Laschi, and M. Calisti, "Morphologically induced stability on an underwater legged robot with a deformable body," *The International Journal of Robotics Research*, 2019.
14. 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.
15. N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *European Conference on Artificial Life*. Springer, 1995, pp. 704–720.
16. S. Koos, J.-B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 122–145, 2013.
17. R. Pfeifer and G. Gómez, "Morphological computation–connecting brain, body, and environment," in *Creating brain-like intelligence*. Springer, 2009, pp. 66–83.
18. G. Hoffman, "Embodied cognition for autonomous interactive robots," *Topics in cognitive science*, vol. 4, no. 4, pp. 759–772, 2012.
19. 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.
20. T. F. Nygaard and J. Nordmoen, "DyRET software repository," https://github.com/dyret-robot/dyret_documentation, 2019.
21. T. F. Nygaard, J. Torresen, and K. Glette, "Multi-objective evolution of fast and stable gaits on a physical quadruped robotic platform," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–8.
22. 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.
23. 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*. Springer, 2018, pp. 719–733.
24. 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.
25. 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.