UiO : University of Oslo

Tønnes Frostad Nygaard

Legging It: An Evolutionary Approach to Morphological Adaptation for a Real-World Quadruped Robot

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Sciences

2020

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 2308

ISSN 1501-7710

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Abstract

Robots are used in increasingly complex environments and need to be able to adapt to changes and unexpected events. This has traditionally been solved by changing the control of a robot, but having an adjustable body can unlock new and powerful adaptive capabilities. An adaptive morphology allows tuning of the physical structure of the robot to different, often conflicting, dynamic requirements, including speed, stability, and efficiency. It can also unlock new functionalities that might not be possible with static morphologies, including variable gearing and multiple locomotion modalities. Even with the potential benefits of morphological adaptation, the methods and technology are still not at a point where there is wide-spread use of adaptive morphologies in physical robots.

The main goal of the thesis is to develop methods and technology to enable adaptation of the physical body of a robot to new real-world environments. An evolutionary approach is taken, and to what degree evolutionary algorithms are able to exploit the dynamic morphology of a legged robot is investigated. The feasibility of continuous adaptation of morphology in realistic outdoor environments is also explored.

A quadruped mammal-inspired robot with the ability to continuously adjust the length of its legs during operation has been designed and implemented as part of the work outlined in the thesis. Evolutionary algorithms are used to optimize both the control and morphology of the robot to different hardware conditions and walking surfaces in the lab. To achieve this, a new gait controller concept with an adjustable complexity is introduced. This allows evolution in scenarios with a wide range of evaluation budgets. A final proof-of-concept implementation of adaptive morphology is also demonstrated. Our robot was shown to be able to adapt its body continuously while walking in different unstructured outdoor terrains, significantly outperforming a non-adaptive approach.

The thesis concludes that adaptation of the physical body of a robot is feasible, and in fact, already shows significant benefits with current technology and methods. Evolutionary algorithms are shown to be effective for adaptation of morphology in a range of different conditions. By developing new methods and technology, as well as demonstrating their utility through real-world experiments, we hope to inspire others to use adaptive morphology on their physical robots.

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* at the University of Oslo.

The research project has been funded by the Research Council of Norway through the project *Engineering Predictability with Embodied Cognition* with project number 240862. The research was conducted at the Robotics and Intelligent Systems research group at the Department of Informatics during the period 2015-2020 under the supervision of Associate Professor Kyrre Glette, Professor Jim Tørresen, and Dr. Charles Martin. In 2019 the author had a six-month stay at The Commonwealth Scientific and Industrial Research Organisation (CSIRO) in Australia, being supervised by Dr. David Howard.

Acknowledgements

I would like to extend my gratitude to my supervisors for all the support I have received during my Ph.D.: Kyrre Glette for keeping me focused on the research while still allowing me to pursue everything needed to build my robot; Charles Martin for serving as a mentor to me and teaching me so much, both in research and beyond; and Jim Tørresen for securing funding and serving as a project leader.

I am very thankful for the opportunity to stay half a year at CSIRO in Brisbane, Australia, under the supervision of David Howard. Seeing all the amazing things they've all accomplished there was truly inspiring. It was the perfect place to bring the robot out of the lab and into realistic outdoor terrains.

I would like to thank my colleagues at the Robotics and Intelligent Systems research group for a great working environment. Especially Jørgen for taking on my robot in simulation and being a sparring partner for all things DyRET; Frank and Benedikte for valuable input, late-night talks, and consumption of questionable beer in various local parks; Kai for interesting discussions on the research project; and Vegard and Mats for sharing my passion for practical robotics. I would also like to thank Jack Collins at CSIRO for helping me getting started Down Under and for carrying me through six months of card game lunches.

Most of all, I would like to thank my girlfriend, Anniken, and the rest of my family and close friends for supporting me through this challenging but rewarding journey. I would not have pursued robotics had it not been for my mother and father, Turi and Ole, always encouraging my interest in computers and technology; and my uncle, Magnus, for the summers he spent helping me with various woodworking and electronics projects. This sparked an interest that I carry with me to this day.

List of publications

Paper I: Multi-objective Evolution of Fast and Stable Gaits on a Physical Quadruped Robotic Platform
T.F. Nygaard, J. Torresen, K. Glette.
The 2016 IEEE Symposium Series on Computational Intelligence (SSCI).
DOI: 10.1109/SSCI.2016.7850167.

Paper II: Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing
T.F. Nygaard, C.P. Martin, J. Torresen, K. Glette
The 2019 International Conference on Robotics and Automation (ICRA).
DOI: 10.1109/ICRA.2019.8793663.

Paper III: Real-world evolution adapts robot morphology and control to hardware limitations
T.F. Nygaard, C.P. Martin, E. Samuelsen, J. Torresen, K. Glette
The 2018 Genetic and Evolutionary Computation Conference (GECCO).
DOI: 10.1145/3205455.3205567.

Paper IV: Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds
T.F. Nygaard, C.P. Martin, J. Torresen, K. Glette
The 2019 European Conference on the Applications of Evolutionary Computation.
DOI: 10.1007/978-3-030-16692-2_41.

Paper V: Environmental Adaptation of Robot Morphology and Control through Real-world Evolution
T.F. Nygaard, C.P. Martin, D. Howard, J. Torresen, K. Glette
Journal paper under review

Paper VI: A Morphologically Adaptive Quadruped Robot in the Wild
T.F. Nygaard, K. Glette, C.P. Martin, Jim Torresen, D. Howard
Journal paper under review

Papers written during the PhD, but not included in the thesis:

Overcoming Initial Convergence in Multi-objective Evolution of Robot Control and Morphology Using a Two-Phase Approach
T.F. Nygaard, E. Samuelsen, K. Glette
The 2017 European Conference on the Applications of Evolutionary Computation
DOI: 10.1007/978-3-319-55849-3_53

Exploring Mechanically Self-Reconfiguring Robots for Autonomous Design
T.F. Nygaard, C.P. Martin, J. Torresen, K. Glette
The 2018 ICRA Workshop on Autonomous Robot Design
arXiv: 1805.02965

Lessons Learned from Real-World Experiments with DyRET: the Dynamic Robot for Embodied Testing
T.F. Nygaard, J. Nordmoen, C.P. Martin, K. Glette
The 2019 ICRA workshop on Learning Legged Locomotion
arXiv: 1905.05626

Experiences from Real-World Evolution with DyRET: Dynamic Robot for Embodied Testing
T.F. Nygaard, J. Nordmoen, K.O. Ellefsen, C.P. Martin, J. Torresen, K. Glette
The 2019 Symposium of the Norwegian AI Society (NAIS19)
DOI: 10.1007/978-3-030-35664-4_6

Evolved embodied phase coordination enables robust quadruped robot locomotion
J. Nordmoen, T.F. Nygaard, K.O. Ellefsen, K. Glette
The 2019 Genetic and Evolutionary Computation Conference (GECCO19)
DOI: 10.1145/3321707.3321762

Understanding Musical Predictions with an Embodied Interface for Musical Machine Learning
C.P. Martin, K. Glette, T.F. Nygaard, J. Torresen
Frontiers in Artificial Intelligence, 2020 Issue 3
DOI: 10.3389/frai.2020.00006

Real World Morphological Evolution is Feasible
T.F. Nygaard, D. Howard, K. Glette
The 2020 GECCO workshop on Evolution of Robots for the Real World
arXiv: 2005.09288

On Restricting Real-Valued Genotypes in Evolutionary Algorithms
J. Nordmoen, T.F. Nygaard, E. Samuelsen, K. Glette
Conference paper under review
arXiv: 2005.09380

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robots inspecting the damaged Fukushima reactor were presented with a daunting task: to pass through a narrow duct to enter the area, traverse gaps between platforms, move over and through various types of debris, and even swim through murky water. Designing a robot to work across such diverse and unstructured environments is challenging, as task and environmental conditions may change, sometimes drastically, during operation. As such, technological limitations meant that the eventual solution required numerous highly specialized traditional robots, with correspondingly high numbers of deployments and extended mission times [37].

An arguably more efficient and attractive solution would be a single 'Swiss army knife' robot. Capable of online morphological adaptation, this robot would be able to match its capabilities to its immediate needs: having at one time a large span to traverse gaps, yet at another time being able to shrink and squeeze through narrow openings in debris fields. Shape-shifting (or morphologically adaptive) robots have long been a mainstay in our collective consciousness[1]. And with good reason; they represent an appealing future where robots have become masters over their environment, able to adopt a variety of configurations to meet their immediate and long-term needs and improve mission outcomes. The underlying principle is that a variable morphology provides additional degrees of freedom to adapt to a given environment compared to a static morphology. This increases the likelihood that the robot can adapt and survive in the face of unpredictable environmental conditions. The challenges presented by Fukushima, chiefly multimodality and unpredictability, are characteristic of the type of unstructured environment that robotic systems as a whole continue to struggle with. In principle, morphologically adaptive robots are a promising enabling technology to unlock a broad swathe of unpredictable environments and solve new tasks on the fly, without having to be redesigned and rebuilt each time they face something unexpected. Due to this promise, morphological adaptation is an area of increasing scientific focus that encompasses a range of research from variable stiffness robot limbs [3] to elegant origami-inspired morphing structures [72] and various soft robots [46].

The field of Evolutionary robotics shows great promise for making use of morphological adaptation [6]. Evolutionary techniques take inspiration from natural evolution and optimize both control and morphology to different tasks and environments [21]. Most work in the field, especially where morphology is evolved, has been focused on virtual robots in simulation [55]. Some transfer some robots to the real world through rapid prototyping techniques, both legged

[1] Often found in science fiction popular culture, e.g., The Transformers, or T-1000 from the Terminator series)

robots [80] and more unconventional configuration [31, 49], but examples where the evolution of morphology is done directly in hardware are rare. The few that do are typically relatively simple robots that often require human intervention or complex external reconfiguration mechanisms [53, 94].

Many challenges need to be addressed on the way to mainstream morphologically adaptive robots. This thesis aims to develop methods and technology that enable a continuous optimization of the body of a legged robot to real-world environments. This includes the development of a robust mammal-inspired quadruped robot that can automatically change the length of its legs during operation. Investigating to what degree an evolutionary approach is able to exploit adaptive morphology exclusively through real-world evaluations efficiently is also asked. Experiments are done in controlled conditions in the lab, as well as in realistic real-world environments outside.

1.1 Research objectives

The main objective of the thesis is defined as follows:

> **Main objective:** Develop methods and technology to enable adaptation of the physical body of a robot to new real world environments.

Many possible robotics tasks could be made easier by adapting the body of a robot. This thesis focuses on basic locomotion to increase the applicability of our findings to the robotics field in general. This is further strengthened by using a capable robot for all proof of concept implementations.

Three research questions are asked to guide the work:

Question 1. How can artificial evolution be used efficiently for a physical mammal-inspired quadruped robot?

Most evolutionary robotics experiments are done on virtual robots in a physics-based simulator. This comes with many advantages, but the inaccuracies in the simulators make the results less applicable for direct transfer to real-world scenarios. Doing evolution in the real world comes with many challenges, many of which are worse when using mammal-inspired quadruped robots. Some of these will have to be pursued during the design and development of the robot platform and gait controller, while others will have to be addressed in the scientific approach and experimental setup.

Question 2. To what degree can we observe the benefits of adapting both morphology and control in real-world evolutionary experiments?

This research question asks whether the evolutionary optimization of morphology and control can be shown to be beneficial under real-world experimentation. First, it is unclear whether a practical implementation can be achieved where the adaptive morphology is complex enough to conform to small variations between different environments while not adding excessive mechanical complexity or requiring more evaluations than feasible in hardware experiments. Secondly, it is unclear whether search algorithms will exploit a dynamic morphology, even if it has a clear advantage for performance. Gaits are often highly adjustable, so small changes to the gait controller can have large effects on the behavior and performance of the robot. Adaptive morphology is often much simpler with fewer dimensions, due to the difficulty of implementing real-world structural adaptation. It can, therefore, be much less responsive to change than control. Algorithms might be too focused on adapting the control to be able to optimize the lesser responsive morphology features simultaneously.

Question 3. How can the physical body of a robot be adapted to new and changing outdoor environments?

Question 3 goes further than the two previous questions and asks how a robot system can adapt its body to new and changing outdoor environments. Since the environments can be both dynamic and unknown, the robot needs to be able to sense the environment as it walks. It also needs to take an online approach, and continuously adapt its morphology. Reconfiguration and evaluation can both take a considerable amount of time to perform on the robot, so a method to intelligently decide which morphologies to test might be required for efficient adaptation.

1.2 Thesis outline

This thesis is a collection of papers. The current chapter gives an introduction to the thesis and the aim of the work. Chapter 2 presents relevant background information and related work. Chapter 3 details the technology and software used and developed in the thesis, while chapter 4 presents a summary of the research conducted. Chapter 5 discusses the findings and puts these into the context of the thesis and the research questions. Avenues of future work are also described. Chapter 6 presents the conclusion. All the papers are available at the end of the thesis. They are based on the approved post-print versions and include minor corrections and changes to adapt them to a standard template for inclusion in the thesis.

Chapter 2

Background

This chapter first describes legged robotics and some of the advantages and challenges associated with them. It then presents evolutionary algorithms, before getting into the application area of Evolutionary Robotics.

2.1 Legged robotics

Legged robots are used extensively for a wide range of applications [88]. They can traverse more challenging terrains than robots with wheels or tracks, and can carry larger payloads while consuming less energy than their flying counterparts. Robots with more legs typically have higher stability and are easier to control, although they do consume more power.

Legged robots operate in different environments, solving different types of tasks and working alongside other robots, animals, or humans. Hard coding solutions for all combinations of possible outcomes are becoming impossible as robots are used in more complex and dynamic environments. Instead, pursuing a system where the robot can learn and adapt to changing environments and unforeseen situations is key as robots are deployed in larger parts of society. Most robots use some form of open-loop controllers, where the robot acts solely based on the controller output. This is referred to as blind walking in legged robotics, which often works well when the robot operates in controlled and straightforward environments. The problem arises when the robot encounters something unexpected. Closed-loop control uses measurements of the robot's state to adjust the command from the controller. An example of a closed-loop approach is the use of active balance, where the sensed pose of the robot and a calculation of its balance point can be used to actively keep the robot in balance, and stop it from falling over. This can be used when walking over rough terrain, or where there is a risk of being shoved [27].

An essential aspect of legged robots when considering locomotion is the notion of stability while walking [18]. The most straightforward approach is to compare the Center of Mass (COM) to its support polygon, defined by all legs in contact with the ground. If the horizontal projection of the COM is within the polygon, the robot is considered statically stable, and will not fall when standing still (without external disturbances). The problem comes from the dynamics of the moving robot, where inertia, friction, and elasticity in effect reduces the size of the support polygon further as the robot moves. This severely limits the speed of statically stable gaits, especially for robots with a significant part of their weight in the legs. Dynamically stable gaits exploit the dynamic effect of the system to stay upright during movement. An excellent example of this is a one-legged hopping robot [81]. It can jump in place and stay upright through advanced control algorithms but will fall if it stops moving. Dynamically stable gaits are more complicated as they require active balancing. It does, however, result in higher achievable speeds and better energy efficiency.

Adapting to the external environment of the robot can be very important, and there are many examples where legged robots [23, 99] successfully adapt to the surface they are operating on. Outdoor environments can change continuously due to weather, which means that even robots operating in a very limited area might need adaptation capabilities. Other aspects that can affect the robot and the need to adapt include changes to its task, as well as other actors in the same environment, like humans, animals, and other robots. Changes to the robot itself can also require adaptation, including wear and tear, as well as mechanical damage [40].

When it comes to terrain interaction in robotics, most work is done using terrain classification [43]. This involves identifying which out of a few predefined terrain classes a new sample belongs to. This can be a powerful technique, but does not capture the variance within each terrain class, and does not determine the actual features of the terrain. Another approach is to do terrain characterization instead [71]. This involves measuring one or more features of the terrain that might affect the robot's performance or behavior more directly. Examples of important terrain features used in robotics are hardness, roughness, slope, and discontinuity [33]. Characterization would, for instance discern the changes an environment might undergo if it started raining, and could give different results on different parts of the same terrain. Both of these techniques can be based on a wide range of different sensing techniques, including vision [33], vibration [95], and haptic feedback [30].

There are many types of legged robots used in research. Six or eight-legged robots are popular for their high stability and ease of control, but they tend to use a lot of power. Biped or uniped robots can be very energy efficient but require very complex control algorithms since they are incapable of statically stable locomotion. Four-legged robots are generally considered a good compromise between efficiency and control complexity, and are used for solving several real-world tasks [96]. Two types of four-legged robots are popular: mammal-inspired robots, and spider-inspired robots, sometimes referred to as sprawling type robots. The difference is in the proximal joints (hips), where this joint in spider robots rotate in the yaw axis, while the joint in mammal-inspired robots rotate in the roll axis [38]. Both types of robots have advantages and disadvantages, but learning to walk is typically considered easier for spider-inspired robots [18]. This is both because they have larger supporting polygons due to the legs being at their sides, but also because they can gradually progress from crawling to walking during the learning or optimization process. When it comes to mammal-inspired quadruped robots, several platforms are widely used in research, including Anymal [34], HyQ [82], and Cheetah [4].

Most legged robots use their control to adapt to new and unexpected events, but some can structurally adapt as well. Changing their body, or morphology, can be a powerful method of adaptation. Some legged robots consist of modules that can be manually reconfigured [28, 87], some even designed specifically for exploration of legged robot morphology [45]. Other legged robots can reconfigure automatically but only have a few discrete morphology states. Examples of this include robots that discard legs if they get damaged [54], and robots that switch to new locomotion modes [47]. Legged robots that are able to adapt their morphology continuously are less common. Some reconfigure the leg stance to continuously change from a prone crawling to upright walking [7], while others change the physical characteristics of the legs to affect the robot's behavior [1, 92, 93].

There exists a wide range of robots with adaptive morphology also outside legged robotics. These include modular robots [13], soft robots [91], swarm robots [12], underwater robots [16], hybrid robots [84], tensegrity robots [78], and origami-inspired robots [72].

2.2 Evolutionary Algorithms

Evolutionary computation is a family of algorithms that takes inspiration from biology. In natural evolution, adaptation occurs through natural selection and random genetic variation. An EA mimics principles by simulating a population of solutions where a selected few individuals can reproduce, and only the "fittest" offspring survive. There are many perspectives and taxonomies used in evolutionary computation, but this thesis takes the approach from Eiben's 'Introduction to Evolutionary Computing' [25].

Figure 2.1 shows a typical implementation of a simple EA. Potential solutions are referred to as individuals, which each contains a number of parameters that the EA will optimize. The algorithm's first step is to initialize the set of individuals, referred to as the population. This can be done through a number of methods, but the most common is to start with random individuals. Their performance is then evaluated. The parent selection mechanism chooses which of the individuals in the population are used for creating new individuals. This is typically done probabilistically, with a higher chance of selecting better performing individuals. The selected parents then undergo a mixture of recombination and mutation. Recombination combines two or more parents into a new individual, while mutation makes small changes to the newly created individuals. These make up the offspring. The parents and offspring then go through a survivor selection mechanism that removes unwanted solutions to keep the population size constant. This is typically done deterministically based on a simple set of rules, in contrast to the stochastic approach in the parent selection stage. This can be based on many different measures, including performance, age, and population diversity. The algorithm keeps generating and testing new solutions until a termination criterion is reached.

The individuals that make up the population can be represented in a wide range of different ways. Solutions within the original problem space are often referred to as the phenotype of the individual, while their encoding that the evolutionary search can work on is referred to as its genotype. Typical data structures used for genotypes include integers and real-typed vectors, bitmaps,

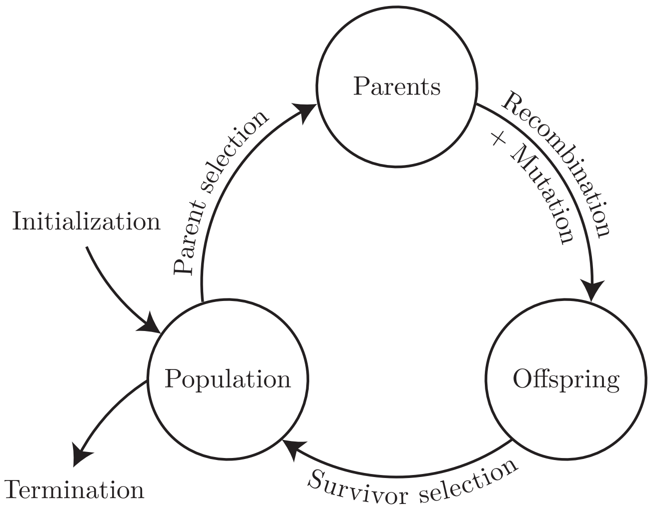


Figure 2.1: The general workings of a simple evolutionary algorithm. A population is initialized, and offspring generated through parent selection and a mix of recombination and mutation. Survivor selection ensures the population size stays constant. The algorithm is looped until a termination criteria is reached.

strings, and different types of graph-based structures. Each element of the genotype is often referred to as a gene, while its value is called an allele.

Another important aspect of an EA is the fitness function. This function evaluates the genotypes and returns their performance, referred to as the individual's fitness. It can be the calculation of a simple mathematical formula, measurement of physical phenomena in an extensive experimental system, or even come from humans evaluating solutions manually. Both minimization and maximization of fitness are typical, depending on the specific application. Many applications only focus on a single objective, while multi-objective optimization involves problems where the performance is measured in several, often conflicting objectives. Several objectives can be combined into one by pre-selecting the desired trade-off between the different objectives, referred to as scalarization [15]. It can then be solved using traditional single-objective approaches. Taking a multi-objective approach can generate a Pareto front that enables choosing the trade-off between the different objectives after the EA has finished. Popular multi-objective evolutionary approaches include NSGA [19, 20], SPEA [103], and PAES [39].

One of the biggest strengths of EAs is its high flexibility. Parent and survivor selection affects the selection pressure and can be tuned to different requirements. Both recombination and mutation have a substantial effect on the diversity of the offspring, affecting how the algorithm moves through the search space. The balance between exploration—testing solutions in new areas of the search space, and exploitation—testing solutions close to known well-performing individuals, can easily be adjusted by either changing evolutionary operator types or parameters. Too much exploration tends to yield inefficient searches with slow convergence, while too much exploitation tends to get stuck in local optima under premature convergence.

Classic evolutionary algorithms focus on improving the performance of the solution. Maximizing the diversity of a population, instead—the variance between the individuals—has, in many cases, outperformed quality-focused approaches [89, 90]. Quality diversity (QD) algorithms have gained popularity in the last few years and aim to provide a diverse set of high-quality solutions [75]. There are many other popular algorithms that can be used instead of, or in combination with evolutionary approaches: Reinforcement learning [9], Bayesian optimization [70], Ant-colony optimization [77], and different types of local search [42], to name a few. These have been shown to be very effective in many cases, while traditional evolutionary algorithms remain a flexible and general solution that can serve as a good starting point for most black-box optimization problems.

2.3 Evolutionary Robotics

The field of Evolutionary Robotics (ER) uses evolutionary computation techniques to optimize different aspects of robots [6]. EAs have been successfully applied in a wide range of settings, from high-level tasks like phototaxis with obstacle avoidance [73] and sequential goal homing [10] to low-level tasks like gait optimization for legged robots [26]. The most common feature to optimize on a robot is the control, but the morphology (body) is also possible. The literature can broadly be separated into four areas, depending on where the optimization takes place: (i) Optimization in simulation alone, (ii) Optimization in simulation with direct transferal to hardware, (iii) Combined optimization in hardware and software, (iv) Optimization exclusively in hardware.

Evolution of control

Designing the control of robots is becoming more and more demanding as both the complexity of the robots themselves as well as the environment they are in and the task they solve increase. Many control problems have traditionally been solved by hard-coding solutions to a few limited expected scenarios and environments, but this is rapidly becoming an infeasible approach.

When it comes to legged robots, considerable effort is being made to optimize the way the robot walks. There are many ways to parameterize the trajectory and timing of a gait [99]. These can roughly be split into two groups: controllers working in joint space and controllers in operational space (or cartesian space). In joint space, the displacement or angles of the joints are controlled directly. This can be achieved through oscillators like Central Pattern Generators [35] other indirect approaches like evolvable Neural Networks [14]. The leg's position in the operational space can also be controlled directly, and the trajectory is often defined with different types of splines [5, 83]. In addition to the movement, there are usually also parameters controlling the timing of the gait. Some controllers have a lot of prior knowledge built-in to the representation, while others are more unconstrained and can generate a more diverse set of gaits. Generally, the less a gait controller is constrained, the more effort is needed for successful optimization.

There is a wide range of different fitness measurements used in ER, with a varying amount of embedded a priori knowledge. When it comes to evolving the gait of legged robots, three of the most common metrics used are speed (or total distance traveled for a set time), stability, and energy efficiency. Optimization can be focused on one of these through single-objective optimization, or more than one of them through multi-objective optimization.

Most work is not done on real-world robots, but on virtual robots in physics-based simulators [55]. The main advantage of this approach is that evaluations are very quick. It is also becoming more accessible as computational power decrease in cost, and experiments can be run on standard desktop machines. There is also no need to design, build, and maintain a robot system, no risk of damage to the robot as it evaluates, and no need for continuous human intervention during evaluations.

Many use simulators to optimize control, but transfer a select few controllers to test on a physical robot. The problem with using a simulator is that they only approximate the real world, and the discrepancy between the virtual robot and the real one can severely limit the usefulness of the evolutionary process. One part of it is that the performance is often not correctly predicted, but since the environment is often entirely static in simulation, the robots have no incentive to develop the robust gaits needed to work in the real world. This inaccuracy is referred to as the reality gap [36], and dealing with it is considered one of the biggest challenges in the ER field [24].

There are several techniques to reduce the reality gap and exploit the simulator's advantages while still getting solutions that work on a physical robot [101]. One of the simplest is to add noise to the simulator to try to add robustness to variance more substantial than the reality gap [36]. Another option is to use local search in hardware to regain some of the performance lost to the reality gap, referred to as memetic evolution [79]. Relying on sensor-feedback to affect control has also been shown to be effective [58]. Self-modeling can be used to continuously update the simulation to changes to the robot or environment [8], and building a map of prior knowledge in simulation can then be used to adapt efficiently in hardware [17]. Modeling the reality gap for different parts of the search space using physical evaluations was also used successfully [41]. The problem is that the best solutions are often the ones that are able to exploit the dynamics of the system and the peculiarities of the environment. These are also the ones that are hardest to simulate and predict correctly, so this approach often removes most of the high performing individuals.

The only way to completely bypass the reality gap problem is to do evolution exclusively in hardware. This has been done both on commercially available legged robots like Aibo [11, 32], as well as custom robots [98, 100]. Some solutions enable autonomous testing of new control in hardware, which alleviates some of the issues of hardware testing by allowing a more extensive evaluation budget [29]. The problem with this is that using simplified environments in the real world still leaves a gap if the robot is not optimized in the environment it will eventually operate in.

Evolution of morphology

Optimizing the body of a robot, its morphology, can be a powerful technique and a key to achieving resilient robots [102]. This is most commonly done as part of the design phase of a system, either as part of a human process or an autonomous design system [22]. Optimizing the robot's morphology has the potential to impact the performance and behavior of the robot in a way that control might not be able to [44, 74]. Combining optimization of control and morphology can, therefore, enable adaptation to a wider range of environments and scenarios than optimizing control alone.

The theory of embodied cognition states that the mind is not the sole source of cognition available to humans and computational systems and that the environment, body, brain, and the interactions between these can all serve as sources of cognition [97]. Changing the morphology of the robot during operation might unlock possibilities to adapt to scenarios and environments not previously considered possible when only changing control. In some cases, changing the robot's morphology might be the only viable option to elicit suitable in-environment behaviors [74]. High-level reasoning often requires little computation power, while low-level sensorimotor skills can require enormous computational resources [52]. Taking an embodied approach to some low-level tasks might free up large amounts of computational resources, possibly to a point where they are now feasible for on-board use in robotics.

One of the seminal works of morphological evolution is Sims' virtual creatures that were optimized for swimming, walking, and jumping in a simple simulator [85]. Most work evolving morphology also optimized control, but there are also examples where the morphology of a robot is evolved alone [48, 86].

The simplest approach for the evolution of morphology for real-world robots is to evolve the design in simulation, then manufacture a select few individuals and verify them in hardware. There are several examples of this being done for legged robots [68, 80], as well as more unconventional robot designs [31, 49]. The challenge with this approach is that the evolved individuals suffer from the effects of the reality gap, which can be even more substantial when both control and morphology are optimized. Combining simulation with hardware experiments is more challenging when the morphology is also optimized. Few robots can morphologically adapt in hardware, and a common approach is to keep optimizing the control alone in hardware [68, 79].

Physical robots with a dynamic morphology are relatively rare in ER. Some can be manually assembled out of modules, either made specifically for evolutionary experiments [2] or based on widely-available LEGO bricks and modules [50]. The problem with these is, of course, that they require human intervention to build and test. The modular approach can be taken a step further and be automated reconfiguration using robot arms [53, 94]. This approach enables more rapid generation and evaluation of individuals, but the robots are relatively limited due to their small size and lack of sophisticated sensor and actuation capabilities. Dynamic morphology has also been successfully used to accelerate evolution and achieve higher robustness and quality of locomotion on legged robots, but the morphology is typically only varied during evolution and not during the final operation of the robot[7].

Chapter 3

Technology and software

This chapter presents the robot build as part of the Ph.D. and the software used for development, analysis, and visualization.

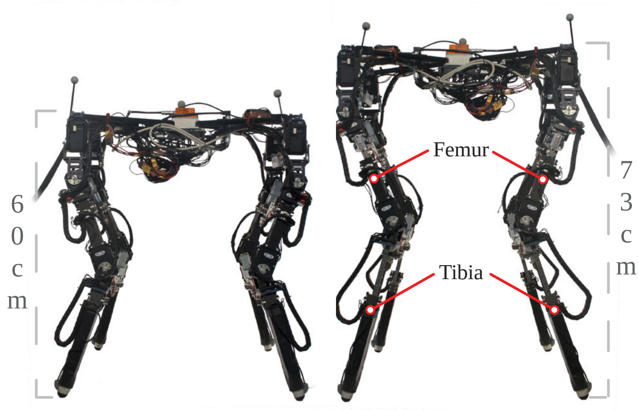


Figure 3.1: The DYnamic Robot for Embodied Testing (DyRET), with its shortest leg configuration to the left, and its longest configuration to the right.

3.1 The DyRET robot

A robust robotic system with adaptive morphology was needed to investigate real-world optimization of robot morphology. No commercial off-the-shelf (COTS) solutions were available at the start of our work, so a custom robot was designed and built from the ground up for use in our research, seen in Figure 3.1.

Design philosophy

Using the robot for machine learning experiments puts tough requirements on its robustness. Many optimization algorithms start with random solutions before gradually improving them throughout the optimization process. This means that many poor solutions will have to be tested and endured by the robot before better solutions are found. This has to be accounted for in the design process.

The robot needs to be able to withstand both poor gaits and morphologies without negatively affecting the body to be useable for prolonged experiments. The robot is designed to minimize the probability of sudden catastrophic failure,

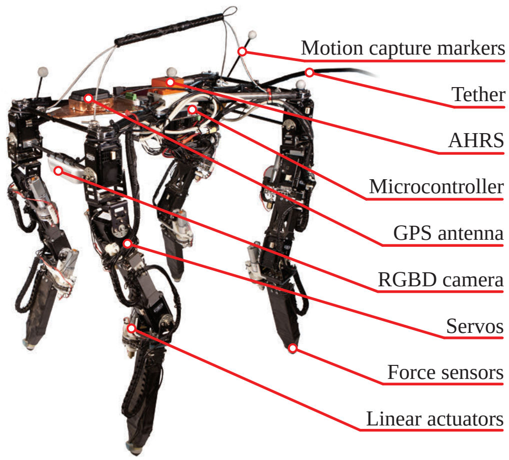


Figure 3.2: The major components of the robot, including sensors, actuators and control.

but that is actually preferable to gradual degradation that affects the behavior and performance of the robot over time. If a replacement part makes the robot perform like it did before it broke, an experiment can be continued. If there has been a gradual degradation, replacing the broken part might drastically change the behavior of the whole system, which might lead to the experiment needing to be started over. Possible failure modes of the different components were therefore carefully considered, and solutions that give the lowest risk of negatively affecting the experimental results were selected.

Keeping weight low is always a challenge when working on legged robots. The most powerful servos available in the Dynamixel MX series from Robotis were used as no other affordable alternatives that suited our requirements were available at the start of our design phase. The closer to the maximum specifications of the servos the joints operate, the quicker they are worn out and the higher probability of permanent mechanical damage. Considerable effort was therefore spent on cutting weight where possible. This was achieved by using light-weight high-strength materials like carbon fiber and aluminum, as well as processes like 3D printing and milling to get custom parts that fit our robot perfectly.

Maintainability and reproducibility are both very important, especially since this is part of a research project with collaborators with different levels of robotics knowledge and experience. Using COTS parts allows easy and quick replacement if they break, although supply might be an issue in the long term. Custom parts can be tailored to the particular needs of the project, and typically perform better, but might require excessive effort to design and produce. For the robot, COTS parts were used where they did not compromise weight or robustness. Custom parts were instead chosen where they had the most significant impact, and enough parts were manufactured to minimize the chance of needing remanufacture for the duration of the Ph.D. project. More details on our design philosophy are available in [66].

Mechanical design

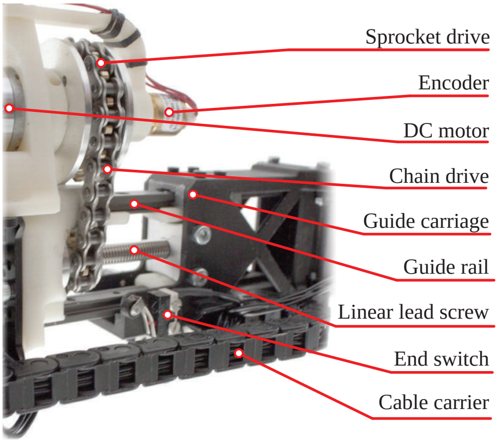


Figure 3.3: The reconfiguration mechanism present in the femur and tibias of each leg.

The robot has a mammal-inspired quadruped joint configuration, seen in Figure 3.1 All mechanical parts can either be bought as relatively inexpensive commercial off-the-shelf components or printed on consumer-grade 3D printers. Some parts can optionally be made in aluminum or other metals for improved robustness. This was done in our case since the robot is used for gait learning experiments. The main body of the robot is constructed with carbon fiber tubing of different diameters, which ensures a stable but low weight base for the four legs. The coxa (hip), femur (top leg section), and tibia (lower leg section) are all revolute joints like traditional mammal robots, consisting of Dynamixel servos from Robotis. The complete robot weighs about 5kg and operates tethered during all experiments.

Each leg also features two custom linear actuators to facilitate changing the leg length during operation, seen in Figure 3.3. The femur has an adaptable length of 50mm, while the tibia can lengthen by 100mm. The longest transition, from minimum to maximum length of the tibia, takes approximately 90s at a speed of about 1mm/s. We chose to use plastic bushings and gliders as much as possible when designing the actuators, replacing traditional ball bearing solutions. Plastic alternatives do not require constant lubrication and have the inherent dirt and dust resistance needed for rough outdoor use. More details on the reconfigurable legs are available in Paper II.

Electrical design and control

The robot can run tethered to a desktop computer and a lab variable 15V bench supply, or from a laptop and a 3-cell lithium-polymer battery for outdoor experiments. The tether features power, USB3 communication, and an antenna cable for the GPS. There is an on-board USB hub for different sensors and peripherals. The servos are controlled through a common serial bus, while the linear actuators are controlled through an Arduino Mega on the underside of the robot body.

Sensors

The robot features a range of sensors to sense its own state and the world around it, seen in Figure 3.2.

Servos — The MX servos from Robotis feature internal sensors measuring current, voltage, and temperature.

AHRS — The robot has an MTI-30 Attitude and Heading Reference System (AHRS) from XSense. It reports linear acceleration, rotational velocity, and orientation at 100hz.

Motion Capture — The robot features four passive infrared reflectors for use with optical motion capture systems. The University of Oslo has an Optitrack system with 14 cameras and coverage of 4x4 meters, while CSIRO has a Qualisys system with 26 cameras and coverage of 8x8m. Both systems achieve a sub-cm accuracy.

GPS — Two c94-m8p differential GPS systems from UBlox are used for outdoor positioning. Real-time kinematic (RTK) positioning is utilized by having a stationary base-station to correct the satellite data received on the robot. This results in a 2d positional accuracy of about one cm.

Depth Camera — A Realsense D435 sensor from Intel is mounted in the front of the robot. The angle is hand-adjustable but has been kept pointing straight down for terrain characterization. It features an RGBD sensor that gives both a standard color image, a depth image, and a point cloud. The rates are configurable but were set to 6hz due to the high volume of data generated.

Foot force — The end of the legs features individual OMD-20-SH-80N force sensors from Optoforce. These report three-axis directional force at 100hz and has been used for terrain characterization. They are easily mounted and dismounted for operation in different environments.

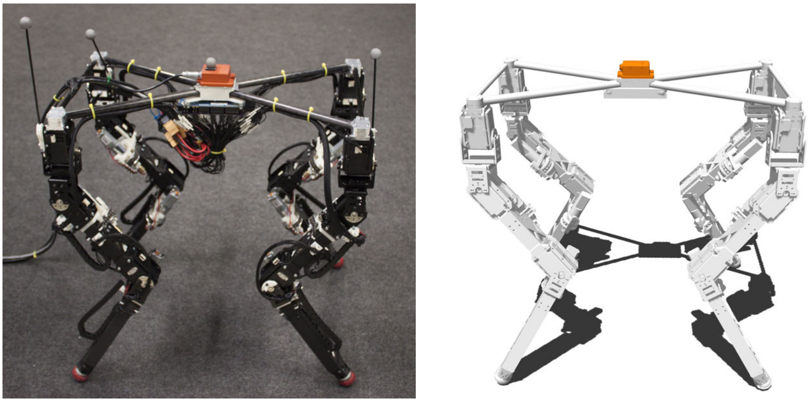


Figure 3.4: The physical robot to the left, with the virtual robot in the Gazebo simulator to the right.

Gait controllers

We have used two different gait controllers in our work. The first is a high-level gait controller with human-understandable parameters. It has a direct encoding, where the shape the leg trajectory is controlled by three parameters: The step height, which controls how high the robot lifts each leg; the step length, which is how long each step is; and a step smoothing parameter, which smooths out the movement as the leg hits the ground, to allow more gentle steps. There is also a frequency parameter that controls the number of steps per second. A balancing wag movement was also added, where the robot leans to the opposite side of the currently lifted leg. Both the phase and amplitude of this movement can be controlled to potentially counteract any dynamic effects as the robot walks. More details are available in Paper I.

We also developed a gait controller with adjustable complexity. The coordinates of the control points in the trajectory spline are represented directly. Timing is controlled through a frequency and lift duration parameter, and the same balancing wag from the high-level controller is implemented. The controller features a dynamic genotype-phenotype mapping that allows the the controller's complexity to be adjusted for different evaluation budgets and environmental requirements. More details available in Paper IV.

Simulation

A virtual version of the DyRET robot is implemented in the Gazebo simulator, seen in Figure 3.4. Gazebo supports the ODE, Bullet, Simbody, and DART physics engines, and is fully integrated with the ROS framework running on the rest of the robot. All actuators are implemented on the virtual robot, and the simulator exposes the same topics and services available on the physical robot. This ensures that all software will be able to work seamlessly with both the

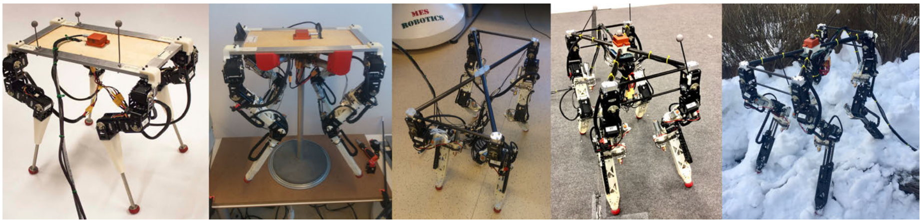


Figure 3.5: A handful of the many iterations the robot has been through during the project.

physical and virtual robot. A custom python server/client job manager has been implemented to enable running simulation jobs across different computers and cloud solutions. More details are available in Paper IV.

Robot iterations

The DyRET robot has been through many iterations during the project. A few selected few can be seen in Figure 3.5.

Paper I: The initial robot design featured a static morphology with reflective markers for motion capture and an AHRS for stability measurements.

Paper II and III: Adaptive morphology was implemented, as well as a redesigned carbon-fiber chassis.

Paper IV: A virtual robot in simulation was developed, along with new, more powerful servos for the femur and tibia for hardware experiments.

Paper V: This paper involved more extensive real-world evolutionary experiments, and many parts were improved for increased robustness and reliability. The reconfigurable length of the femur was doubled to allow a broader range of morphologies. New silicone toes were also added to handle different walking surfaces better.

Paper VI: Both an RGBD-camera in the front and force sensors in all four feet were added for terrain measurements. The USB system was upgraded to USB3 to support the higher bandwidth needed for the depth camera. A GPS antenna was also added to enable precise outdoor positioning.

3.2 Software

The whole software system uses the Robot Operating System (ROS) framework [76]. Low-level control code is written in C++, while some higher-level functions and all visualization and analysis code is written in Python.

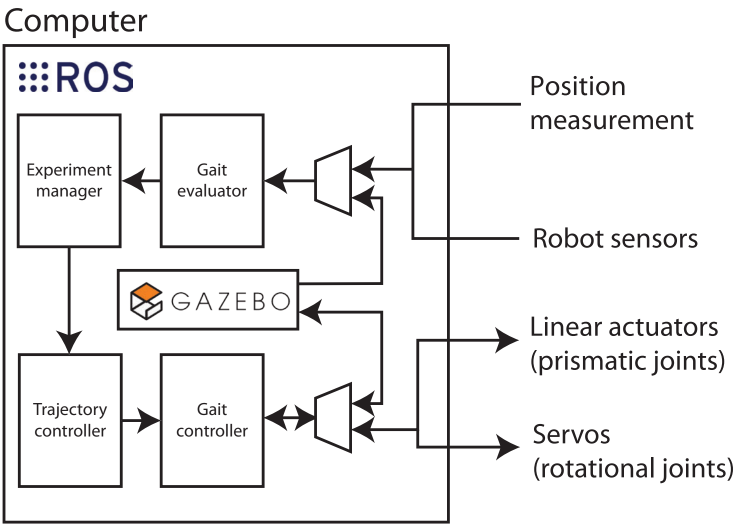


Figure 3.6: Overview of the software system for the robot.

Figure 3.6 shows a system overview. We have an experiment manager node that oversees everything and runs the specific code necessary to start and track the experiments. It sends walking trajectories to the trajectory controller, which in turn selects a specific gait to use to achieve that trajectory. This node interfaces either to the physical robot or to a virtual robot in the Gazebo simulator. A gait evaluator node evaluates the performance and behavior of the current gait and morphology, based on position and sensor data received from the simulator or physical robot. All data is also sent back to the experiment manager for logging.

Other frameworks and software used

This is a short list of the software and frameworks used in the thesis. This list is primarily included as a reference for anyone doing related work.

ROS	Robot Operating System (ROS) is a framework for collaborative robotics software development. We use C++ and Python for our custom code, and interface to many other packages.
Gazebo	The Gazebo simulator interfaces to ROS, and includes several physics engines like ODE, Bullet, Simbody and DART.
SplineLibrary	The SplineLibrary by Elliott Mahler is a powerful C++ framework for generating and manipulating splines of different types.
Sferes2	A template-based C++ framework for evolutionary computation.

RabbitMQ	Open source message-broker software that enables abstract message passing over the network through libraries in all major programming languages.
PCL	The Point Cloud Library (PCL) is an open source library for point cloud and 3D geometry processing, written in C++.
OpenCV	Cross-platform framework for real-time computer vision functions.
Jupyter notebook	Web-based interactive computational environment for python programming, used for analysis and visualization of data.
Seaborn	Python visualization library based on matplotlib.
Overleaf	Online latex collaborative writing and publishing tool used for papers and the thesis.
Illustrator	Vector graphics editor from Adobe, used for production of illustrations and diagrams.
DaVinci Resolve	Color correction and non-linear video editing program from Blackmagic design.
Adobe Audition	Digital audio workstation from Adobe, used for recording voice-overs for videos.
Clion	Cross-platform IDE for C and C++ from JetBrains, used for low level coding.
PyCharm	Cross-platform IDE for python from JetBrains, used for scripting and programming of sensor and modeling code.
Fusion360	CAD and CAM package for design, simulation and manufacture of 3d parts.

Chapter 4

Summary of papers

This chapter provides summaries of the papers included in the thesis. Section 4.1 gives a high-level overview of how the papers relate to the research questions, while Section 4.2 goes into detail on each individual paper.

4.1 Overview

The six papers included in this thesis iteratively address the three research questions, and also detail the design and development of the robotic system needed to perform the experiments.

Figure 4.1 shows how the papers are connected to the research questions. Paper I describes our initial approach to answering the first research question before we started investigating the second question in Paper II and III. We decided to revisit parts of the first question in paper IV, after seeing that our previous solution was not optimal. After improving our controller, we were able to wrap up our investigation of the second question in paper V. Finally, paper VI describes our work on the third and last research question.

Table 4.1 gives a short overview of the aim, methods, and results of each paper.

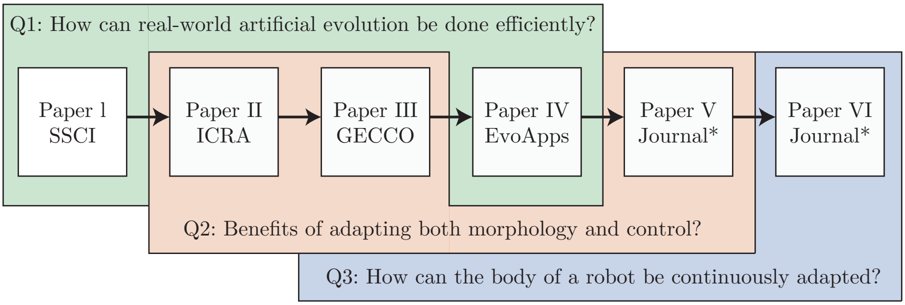


Figure 4.1: All the papers included in the thesis, grouped by the research questions they address. Please see Section 1.1 for the full research questions. *These journal papers are under review at the time of thesis submission.

Table 4.1: Overview of the aims, methods, and results for the papers in the thesis. The last two papers are journal papers under review at the time of thesis submission.

	Aim	Methods	Results
I SSCI	• Investigate feasibility of multi-objective optimization in hardware alone. • Design and build robot platform. • Develop a suitable high-level gait controller.	• Comparison of single-objective and multi-objective optimization of control in hardware, with speed and stability as fitness measures.	• Demonstration of real-world multi-objective evolution. • New high-level gait controller. • A single-objective approach is not sufficient.
II ICRA	• Implement adaptive morphology on our robot platform. • Investigate potential benefits of adaptive morphology.	• Lab experiments under different hardware conditions. • Outdoor experiments in different environments.	• Proof of concept implementation of a mechanically adaptive robot. • Demonstration of the potential benefits of adaptive morphology.
III GECCO	• Investigate adaptation to hardware limitations using evolutionary algorithms.	• Multi-objective co-evolution of morphology and control with differences in available servo torque.	• Demonstration of evolutionary adaptation to hardware limitations. • Evolution utilizes both control and morphology.
IV EVOAPPS	• Develop a new gait controller with adjustable complexity. • Investigate interaction between controller complexity and evolvability.	• Comprehensive multi-objective evolutionary runs in simulation. • Limited multi-objective evolutionary runs in hardware.	• Development of a new gait controller approach with proof of concept implementation. • Demonstration of interactions between controller complexity and search performance.
V	• Investigate adaptation to different walking surfaces through evolutionary techniques.	• Multi-objective co-evolution of morphology and control on different carpets in the lab.	• Demonstration of evolutionary adaptation to different walking surfaces.
VI	• Develop method for continuous adaptation to unstructured terrains. • Comparison of adaptive and non-adaptive approaches to walking on unstructured terrains.	• Modeling of how terrain affects efficiency for different morphologies. • Adapting to previously seen homogeneous terrains in the lab. • Adapting to previously unseen heterogeneous terrains outside.	• Proof of concept implementation of continuous morphological adaptation on real-world terrains. • Demonstration of the benefits from morphological adaptation under real-world conditions

4.2 Papers

This section gives the motivation behind each individual paper, in addition to a summary. The full papers are available at the end of the thesis.

4.2.1 Paper I

Multi-objective Evolution of Fast and Stable Gaits on a Physical Quadruped Robotic Platform
Tønnes F. Nygaard, Jim Torresen, Kyrre Glette
In *2016 IEEE Symposium Series on Computational Intelligence*

This first paper lays the foundation for doing real-world evolutionary experiments. This is done by developing a high-level gait controller and robotic platform that enables multi-objective evolution on a physical quadruped mammal-inspired robot without aid from physics simulators.

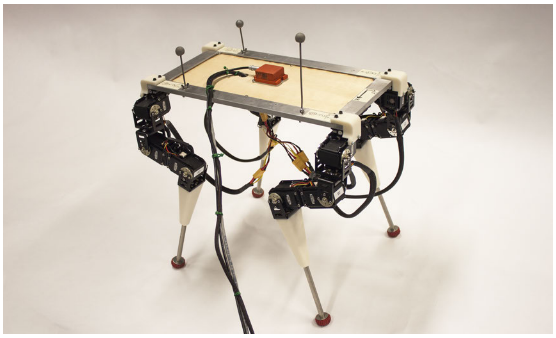


Figure 4.2: The first iteration of the quadruped robot, used in Paper I.

Motivation

There are very few examples of real-world evolutionary robotics research being done on physical legged robots. When looking at mammal-inspired quadruped robots, there are even less, and most only employ short single-objective runs. Being able to do larger multi-objective evolutionary runs under the strict evaluation budgets in hardware would allow evolutionary robotics techniques to be used for a much broader range of applications.

Summary

Doing evolution directly on a physical legged robot is very challenging. One of the biggest issues is that each evaluation can take a considerable amount of time, which severely limits the available evaluation budget. The physical robot is also prone to gradual degradation and catastrophic mechanical failure.

To enable evolution under a limited evaluation budget, we implement a new high-level gait controller that produces a continuous, regular wave gait. Seven human-understandable parameters shape the trajectory of each leg. We demonstrate this new controller in both single-objective and multi-objective approaches to evaluate its feasibility for hardware-only evolution.

Our results show that single-objective evolution is insufficient when using speed and stability as fitness measures. Evolving only for speed results in individuals with a very high probability of falling. Evolving only for stability results in individuals that move too slow for practical use. Multi-objective evolution, however, yields solutions with a wide range of different trade-offs between the two objectives. We also demonstrated that our new high-level gait controller was a good fit for our limited evaluation budget.

The high-level gait controller is used directly in Paper II and Paper III, before being replaced in Paper IV.

4.2.2 Paper II

Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing
Tønnes F. Nygaard, Charles P. Martin, Jim Torresen, Kyrre Glette
In *2019 IEEE International Conference on Robotics and Automation*

In this paper, we present our robot's new adaptive morphology. We demonstrate the potential benefit of morphological adaptation by testing two hand-picked body shapes under different conditions, both in the lab and in outdoor environments.

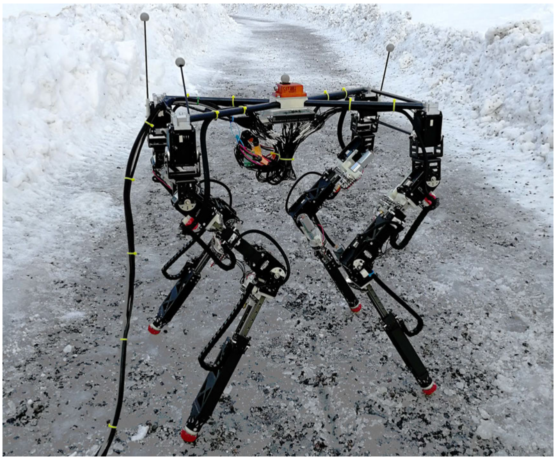


Figure 4.3: The second iteration of the robot, with dynamic morphology. This image was taken during one of the outdoor experiments in Norwegian winter conditions.

Motivation

Allowing a robot to change its own body during operation increases its adaptive power, potentially allowing higher performance and the ability to adapt to a broader range of environments and tasks. It is not self-evident how this could be implemented on a mammal-inspired quadruped robot like ours. It is also unclear to what degree having a dynamic morphology for this type of robot would actually be beneficial, and if it would improve performance over traditional static morphologies.

Summary

Choosing how much and which parts of a robot's body is made adaptable will severely affect the impact on behavior and performance. The choice often involves a trade-off between how effective the change is on one side, and increased weight and mechanical complexity on the other. It can also be hard to know what effect the morphological adaptability will have before it is implemented and tested on the actual robot, and to what degree this will be useful in the environment it will eventually operate in.

In the paper, we present our shape-shifting robot platform with the ability to change the length of its legs during operation. We introduced our hypothesis: *No single robot morphology performs best for all situations, tasks, or environments*. To test this, we hand-picked two dissimilar body shapes to test, one with the shortest possible leg length, and one with significantly longer legs. We tested them under different conditions, both in the lab and outside. Inside, we tested the two morphologies with different supply voltages to the actuators. Lower supply voltage leads to a reduction in available joint torque of about 20%, which makes it much harder for the robot to walk. Outside, we first tested the robot on a flat concrete surface in a covered garage, before taking it outside on an icy footpath that was considered a much more challenging environment for the robot.

We saw that the tall robot was able to achieve higher speed in the lab conditions than the one with short legs, but this flipped once the voltage was reduced, where the short-legged robot now outperformed the longer one. Outside, we saw that the tall robot performed best in the garage environment, while the short-legged robot walked fastest on the icy path. This corresponded well to the results indoors. In both experiments, we saw that longer legs were better in the less demanding environments, while shorter legs worked better for the more challenging ones.

This is the robot design that is used for the remainder of the papers in the thesis, with small modifications for increasing robustness and adaptability.

4.2.3 Paper III

Real-World Evolution Adapts Robot Morphology and Control to Hardware Limitations
Tønnes F. Nygaard, Charles P. Martin, Eivind Samuelsen, Jim Torresen, Kyrre Glette
In *2018 Proceedings of the Genetic and Evolutionary Computation Conference*

This paper demonstrates morphological optimization to hardware limitations through the use of evolutionary algorithms based exclusively on real-world evaluations. Evolution is able to exploit both the control and morphology when adapting to the reduced actuator torque.

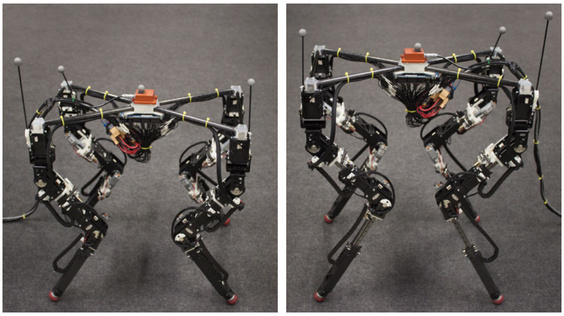


Figure 4.4: Comparison of the shortest and longest legged individuals used for evolution.

Motivation

Artificial evolution is a good candidate for optimizing robot morphology. In the field of Evolutionary Robotics, however, this has almost exclusively been done in simulation, which suffers from the reality gap when the results are transferred to the real world. Evolving the body shape directly in the real world bypasses these issues, and allows adaptation to intricate and rich environments that can't be replicated in simulation.

Summary

There is no guarantee that evolution is able to exploit a shape-shifting robot, especially in a real-world experiment with a limited evaluation budget. Optimizing the gait controller is an easier task, as changes have a more direct effect on the behavior and performance of the robot, meaning that optimization might not be able to focus on morphology at all when both are being optimized at the same time.

In this paper, we describe multi-objective evolution of control and morphology under different hardware conditions. The servo's voltage is varied between different evolutionary runs, reducing the available actuator torque by about 20%. After the evolutionary runs, we also hand-pick some individuals evolved at the higher voltage and re-test them under lower voltage. This gives us a baseline of how lowering torque affects individuals and allows us to better analyze the differences in the individuals evolved under different conditions.

When reducing the voltage on a few hand-selected individuals, we saw significant reductions in fitness for all but the slowest individuals. Evolution was able to adapt to the decreased torque and achieve similar performance on individuals of low and medium speeds. We found statistically significant differences in the two populations for both control and morphology, showing that evolution utilizes both to achieve the results.

4.2.4 Paper IV

Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds
Tønnes F. Nygaard, Charles P. Martin, Jim Torresen, Kyrre Glette
In *2019 European Conference on the Applications of Evolutionary Computation*

In this paper, we introduce our new adjustable complexity controller concept. We motivate and demonstrate the concept through simulation and real-world experiments. Being able to adjust the complexity of a controller allows the same controller to be used for both simulation and hardware evaluation budgets, as well as a range of different environments with distinctive requirements.

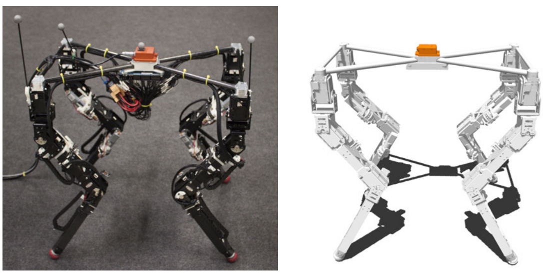


Figure 4.5: The physical robot is seen to the left, with its virtual counterpart in the Gazebo simulator to the right.

Motivation

Choosing the right gait controller complexity can be quite challenging. Simple gait controllers do not require as much effort to optimize, but also have less flexibility. This limits the range of possible gaits, which might not allow enough distinct gait types to be useful for a wide range of environments or tasks. More complex gait controllers are able to produce a broader range of gaits that work well in more situations, but they also require substantially more effort to optimize. Hand-selecting the controller complexity early in the development process can be challenging, especially with changing needs.

Summary

Doing evolution in simulation and hardware allows very different evaluation budgets. In hardware, each evaluation can take a long time, sporadic human intervention is often needed, and there is always a chance of mechanical failure. Therefore, real-world evolution typically has low evaluation budgets that limit the complexity of controllers and morphology. The evaluations in simulations are most often fully automated, and additional computational resources are now so cheap that immense evaluation budgets are possible. The complexity of a controller, which directly affects the effort needed to optimize and the budget needed to do so successfully, is typically chosen while designing and implementing a gait controller. This means that a trade-off must be chosen early in the design process, potentially limiting the types of experiments possible later on.

We propose a new gait controller concept with a dynamic genotype-phenotype mapping. The controller complexity can be tuned by adjusting a single parameter. The complexity can be reduced when hardware experiments demand it or increased for simulation-based experiments with cheaper evaluation mechanisms. It can also easily be tuned to the requirements of the environment or task the robot is solving, which can conserve valuable evaluation time.

We test various complexity parameters in simulation and show through extensive evolutionary experiments that it conforms to our expectations regarding different evaluation budgets. We also verify our results with experiments in the real world, which supports the findings in simulation.

The controller concept is used for the last two papers of the thesis.

4.2.5 Paper V

Environmental Adaptation of Robot Morphology and Control through Real-world Evolution
Tønnes F. Nygaard, Charles P. Martin, David Howard, Jim Torresen, Kyrre Glette
In review (journal)

We demonstrate evolutionary optimization of morphology and control in different real-world environments. The evolved individuals are also tested on other surfaces, and show more similar performance on quantitatively similar terrains, supporting the notion that this technique can be generalizable to a wide range of environments.

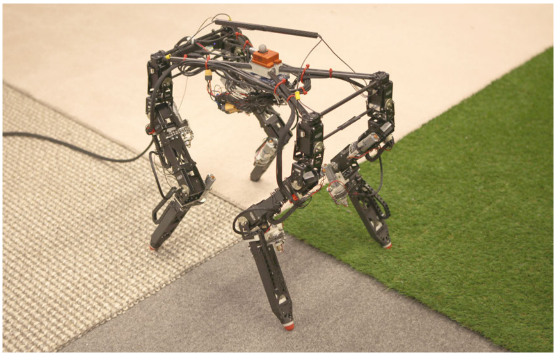


Figure 4.6: The robot on the four different surfaces used in the experiments. The robot was evolved on the top and bottom carpet, while the left and right was used as control for generalization tests.

Motivation

Robots operate in more and more complex environments, and hard-coding how the robot should react to every imaginable situation it might encounter is impossible. One of the main advantages of legged robots is their ability to traverse difficult terrains, but it is essential for the robot to have the ability to adapt to its surroundings for it to do so efficiently. This is traditionally done by adapting the gait of the robot, but having a body that can change could give the robot even more options when adjusting itself to the demands and peculiarities of new and changing terrains.

Summary

Successfully demonstrating environmental adaptation through real-world evolution is challenging. The controller and morphology have to be reasonably complex to be able to adapt to minute differences in walking surfaces, which requires large evaluation budgets. The more evaluations that are needed, however, the higher the chances of gradual degradation or catastrophic mechanical damage invalidating the results of the experiments.

We do multi-objective evolution of morphology and control on two different carpets with different hardness. We do multi-objective evolution on both surfaces, and evaluate and compare the resulting populations. We also evaluate some of the evolved individuals on two new surfaces with a different texture than the original surfaces, to investigate to what degree the individuals found during evolution will generalize on qualitatively similar terrains.

The evolved individuals showed significantly different control and morphology for evolution on the two carpets. The evolved individuals showed more similar performance when walking on qualitatively similar surfaces to the ones there were evolved on, suggesting the technique might be effective for adaptation to quantitatively similar outdoor environments as well.

4.2.6 Paper VI

A Morphologically Adaptive Quadruped Robot in the Wild
Tønnes F. Nygaard, Kyrre Glette, Charles P. Martin, Jim Torresen, David Howard
In review (journal)

In this paper, we demonstrate a system that is able to continuously and efficiently adapt the morphology of our robot to real-world unstructured terrains. The fact that this can be done quickly and with only a minimal baseline dataset suggests adaptive morphology could be an effective technique with many practical applications.

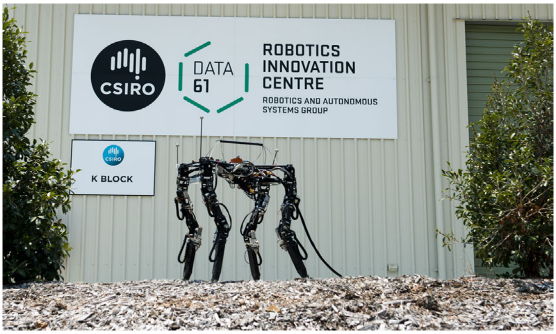


Figure 4.7: The robot at the Australian site used for all the experiments in the paper.

Motivation

Morphological adaptation of legged robots is a powerful technique, and being able to do so in the robot's operating environment directly ensures a good fit. There are two significant challenges to achieving this. The first is that natural environments are rarely homogeneous, and might change significantly, even over short distances. The second is that real-world terrains change quickly and often. It can be the effect of weather conditions like rain, wind and changing temperatures, or external disturbances from humans, vehicles, or other robots. For a robot to be able to operate efficiently in these types of environments, they need to adapt continuously during operation.

Summary

Adapting a robot's morphology in the real world is very challenging, especially when it is to be done continuously. Switching between morphologies on our robot can take up to a minute and a half, so testing all possible morphologies each time the terrain changes is impossible. A baseline model allows bootstrapping of the problem and lets the robot intelligently search for the best morphology for the environment while still learning during operation.

Large indoor boxes with different terrain materials are used to let the robot build a baseline dataset of how different morphologies perform on various types of terrains. A simple adaptation algorithm is implemented and tested on previously seen homogeneous terrains in the lab. The robot is then taken outside, where it uses an extended adaptation algorithm on previously unseen heterogeneous terrains. The adaptation approach is compared to selecting the best performing all-rounder from the terrains inside.

We show that the system works well inside, where the robot is able to utilize the baseline model to efficiently change body shapes when needed. This ensures that energy efficiency is kept high, even for terrain transitions that require very different body shapes. When we run the robot outside on previously unseen heterogeneous terrains, we see that it is able to continuously optimize the body while walking. It does so more efficiently on terrains similar to the baseline dataset, but it is also able to learn and retain the knowledge on new terrains it encounters during operation. The adaptation algorithm is shown to outperform the best all-round morphology from the baseline dataset.

Chapter 5

Discussion

This chapter starts with looking at the contribution of the papers in relation to the original research questions of the thesis, before describing the approach and the design choices made. Limitations are then discussed, before describing avenues for future research.

5.1 Research questions

Question 1. How can artificial evolution be used efficiently for a physical mammal-inspired quadruped robot?

The main difficulty when it comes to doing evolution in hardware on a physical robot is to tune the process to the low evaluation budget. Restricting the size of the search space allows the evolutionary process to converge in fewer evaluations. Limiting it too much, however, will make the robot unable to exhibit the full range of behaviors required to tailor to more than a very narrow range of environments.

In Paper I, this was approached by hand-designing a high-level gait controller with human-understandable parameters. It was shown to be well suited for multi-objective evolution in hardware, and we showed that the evolutionary search was able to generate solutions with a wide range of trade-offs between speed and stability in a reasonable amount of time.

We realized that our controller was more limited than we initially thought during work on Paper III. It was well suited for the most straightforward environments in the real world. The problem came in the more complex environments that require a more exotic walking pattern. This was simply not possible to represent with the chosen high-level parameters. We also saw that the controller was unsuited for situations where the evaluation budget was much higher, like what is usually the case in simulation. The search converged after only a minute or two of simulation, which does not take advantage of the computational resources available—one of the main advantages of simulation.

In Paper IV, we present a new idea where the complexity of the controller itself can be adjusted. We use a dynamic phenotype-genotype mapping that is scaled by a single complexity parameter. This means that the search space can be kept smaller for situations where we either have a minimal evaluation budget or an environment that does not need more than a simple gait. For more demanding environments, or where we have larger evaluation budgets, the search space can be increased to include a broader range of gaits.

We have demonstrated that real-world evolution can be done efficiently on our mammal-inspired robot through these two papers. The key to doing this is controlling the difficulty of the search through the gait controller complexity.

We demonstrated that this could be done successfully by hand for simple environments, but also showed a more general solution where the complexity can be adjusted on-the-fly to suit a wide range of evaluation budgets or environmental requirements. Utilizing simulations in combination with real-world experiments is not needed to achieve effective optimization.

Question 2. To what degree can we observe the benefits of adapting both morphology and control in real-world evolutionary experiments?

We started working on this question in Paper II, where we hand-picked two dissimilar body shapes to evaluate in different environments. We believed the longer legs would do better in simple and less demanding environments, as the additional length allows the robot to take longer steps with the same rotational velocity of the servos. The shorter legs cannot match the step length, but will in turn have the higher force surplus needed in demanding environments. Both indoor and outdoor experiments supported this, which implies that the general notion of adjustable leg lengths could be beneficial in adjusting to different operating environments.

In the next two papers, we used artificial evolution to optimize both control and morphology. Paper III looks at how evolution adjusts to a loss of actuator torque of about 20%, and we saw that evolution allowed the robot to retain its performance for low and medium speeds after the reduction. Paper V investigates how evolution would adjust to changes in the walking surface. We found that the search was able to adapt the robot to the different surfaces and that the solutions found would generalize to other previously unseen terrains as well. In these two papers, we saw significant differences in both control and morphological parameters between the different populations. This shows that evolution can exploit the adaptable morphology, and also serves as a demonstration of the benefits of a shape-shifting robot.

Question 3. How can the physical body of a robot be adapted to new and changing outdoor environments?

This question was addressed in Paper VI. Outdoor terrains are unstructured and change over time, so an approach that successfully adapts the robot's morphology in these types of terrain needs to do so continuously. There are two significant challenges to achieving this. The first is that adjusting the robot's morphology often takes much time, in our case, up to a minute and a half for the most extreme changes. Evaluating locomotion in unstructured terrains will also be very noisy, so getting an accurate assessment of performance makes the process even more time-consuming. The second is that terrains can be very dynamic, and change considerably from step to step. The combination of these two makes it impossible to test all possible morphologies for each terrain change detected. A model-based approach allows the robot to select which morphologies to test based on previous experience, making the process feasible on a physical robot. In our case, we made an adaptation algorithm that transitions between the most energy-efficient morphological configurations based on a model and the terrain it senses. The model is built from experiments in controlled conditions, and we test the system outside on previously unseen and unstructured terrains. We also compared performance to choosing the best all-round static morphology and found that continuous adaptation outperformed it on all the terrains we tested. As technology progress, we expect the cost of onboard mechanical reconfiguration to reduce, and as methods and technology develop further, we expect the efficiency and therefore reward of reconfigurability to increase. The fact that the robot in our very simple proof of concept study performed so well gives strong evidence that efficient model-based continuous adaptation of the physical body of a robot can not only be feasible but highly beneficial in changing real-world environments.

5.2 Approaches and design choices

Many types of robots could be used in our research. Legged, wheeled, underwater, flying, soft, and various types of hybrid robots can all benefit from adaptive morphology. We chose to pursue legged robots for their interesting possibilities and challenges and widespread use in ER. They have been praised for their potential ability to traverse rough terrains, though only recently have successful commercial products with advanced capabilities been made available. We wanted to focus on applicable real-world robots, and legged robots seemed to be in the right stage of development and adoption to give us the highest degree of impact from our work. Many types of legged robots could be used for our research, but we chose to use a mammal-inspired quadruped design, which gives a good trade-off between efficiency and control complexity. At the start of the project, no commercial off-the-shelf robot was available with dynamic morphology. Modifying already existing robot platforms would not give the level of integration we wanted to achieve for the dynamic morphology, and it was not deemed to give a better return-on-investment in terms of time and effort than building a new robot. We therefore started the design and development of a new mammal-inspired quadruped robot with built-in dynamic morphology. This has taken a considerable amount of time and effort throughout the project, but doing the long experiments we have done in hardware would not be possible without a custom robot.

Robots are used for a wide range of tasks, even within the field of Evolutionary Robotics. We chose to look at basic locomotion, specifically walking gait and morphology optimization. Optimizing or adapting the walking pattern of legged robots is a challenging undertaking, especially when considering real-world environments. Classic gait optimization approaches based on mathematics and physics have successfully been used to allow a legged robot to traverse rough and unstructured terrains, but these are typically tailored to specific robots and the environments they operate in. Using biologically inspired methods to give the robot the ability to learn how to walk lets it adapt to the environment it operates in and the task it tries to perform, even as these might vary. Damage and unexpected changes to the robot can also disrupt gaits developed through traditional approaches, while a robot capable of learning has a chance to find alternative walking methods that might not be as affected. Even looking specifically at legged robot locomotion, there are many different performance measures used. In Paper I, we tried single-objective optimization for both speed and stability, respectively, but found that these needed to be used in conjunction to get practically applicable gaits. This was also used for subsequent papers, where we optimized both control and morphology. In our last paper, we only optimized the body of the robot while the control stayed the same. We expect to see much less of a difference in the speed since the controller is not evolved, and we considered stability to be too noisy to give a good measure of the quality of different morphologies when used in unstructured terrain outside. Instead, we decided to use the energy efficiency of the robot, based on the cost of transport calculated using the power draw from the servos. There are many alternatives to both the type and implementation of our fitness measures, but our choices were considered suitable for our needs and resulted in successful adaptation in both indoor and outdoor environments.

Adaptive morphology can be implemented in many different parts of the robot. Since our focus was on locomotion, the goal was to maximize impact on the system's dynamics in a way that would affect the behavior while walking. This has to be achieved without adding too much weight or mechanical complexity to the robot. Several different approaches were considered, but it was ultimately decided on varying the length of the legs. This was considered feasible with current technology while still allowing the robot to have a satisfactory degree of flexibility to impact the behavior and performance. It turned out to be an excellent choice, as evolution was successfully able to evolve it alongside control, while the simple adaptation algorithm significantly outperformed a non-adaptive approach in unstructured outdoor terrains.

We chose to do almost all our work purely in hardware. Many research groups are working on reducing the reality gap using different approaches combining simulations and hardware experiments. We saw that trying to investigate morphological evolution in hardware while at the same time relying on simulation could quickly draw too much focus away from our main goal of morphological adaptation in the real world, requiring us to do extensive work on reducing the reality gap instead. Many techniques can reduce the reality gap to manageable levels for simple environments but were not considered sufficient in our case, as we wanted the robot to demonstrate morphological adaptation in changing outdoor environments in the real would.

5.3 Limitations and future work

Dynamic morphology was implemented on the robot through variable femur and tibia lengths. This gave us enough flexibility to affect the robot's behavior to a sufficient degree, but it is still considered a relatively simple morphological parametrization. We have been focusing on working with robots relevant to solving real-world problems, and part of this was minimizing any reduction in the robot's ability to solve other tasks while carry heavy payloads and other on-board sensors. This put limitations on the degree of morphological adaptation we could add, and a different approach could allow a much more flexible structure. Another rotational joint above the hip could, for instance, allow transitions between mammal and spider-inspired morphologies. We have only looked at adaptive morphology affecting the locomotion of the robot, but adapting sensor morphology could also be interesting, and an embodied solution might improve terrain adaptation capabilities. We also take a very traditional mechanical single-robot approach to adaptive morphology. Many approaches are rapidly becoming more mature, and equally capable self-adaptive robots from modular, swarm, or soft robotics might soon be a reality. New and exotic shape-shifting materials can be key to enabling more dramatic morphological variation and reduced mechanical complexity.

By doing everything in the real world, we have a very limited evaluation budget. This, in turn, makes us unable to use very complex representations of control and morphology that would require too many evaluations to optimize on the physical robot. Even though the results we do get are very relevant for solving real-world tasks, we have not been able to investigate the underlying effects to the same degree someone doing experiments in simulation would be able to. We have, however, been able to verify our methods and techniques in realistic real-world environments, which we believe increases the relevance and impact of our work considerably. Even though optimizing the robot in its final operating environment is the only way to bypass the reality gap completely, there is much useful information that can be extracted from a virtual robot. There are already many methods of combining simulation with real-world experiments. Having a morphologically adaptive robot capable of the type of long experiments we have done in hardware might benefit from different approaches than previously used, where simulation has typically been the focus with small experiments with a handful of real-world evaluations.

Our focus has been to use a highly capable robot with applicability to other applications in robotics to increase the impact of our work and encourage the adoption of adaptive morphology in new areas of robotics. The issue with this is that a capable robot requires considerable more effort to design and build, when compared to a more simplistic and minimal robot. Other approaches in modular and swarm robotics have been able to utilize parallel evaluations in hardware with several physical robots, but this is not possible with a robot like ours. Fully autonomous test setups are also much more challenging to make for legged robots, especially for mammal-inspired configurations. Due to this, all our experiments have required some degree of human intervention, again limiting the extent of our experiments. Further developing the platform and the supporting experimental infrastructure around it could allow a fully autonomous test rig, which would enable much larger evaluation budgets. This could allow both longer evolutionary runs to solve more complex problems and a higher number of runs that could give higher statistical confidence in the results. Removing the reliance on a human operator might also enable external researchers to use the platform remotely.

We chose to only look at legged robot locomotion, and specifically the gait and morphology when walking straight forwards. This is a very low-level problem and can be hard to combine with other high-level concepts or tasks as the robot only walks straight forward. The robot, however, already features a range of different sensors and can carry payloads needed to solve more complex tasks. Adaptive morphology might give an edge in different high-level tasks as well, and our robot could be a possible candidate for use in these types of experiments.

Different gait controllers are used in our papers, but they all take an open-loop approach. The actual state of the legs and the body are not sent back to the controller, so no adjustment is made at the controller level. This means that the controller cannot react to the environment by tuning its output, and adaptation to new environments can only be made by optimizing of the parameters to the gait controller. Allowing closed-loop control would potentially enable more fine-grained and reactive adaptation, and evolution could still optimize both high-level features and the parameters of the reactive control.

Only morphology was optimized when doing continuous adaptation in the real world. An important aspect of that part of the work was getting an accurate measure of terrain characteristics to guide the algorithm in selecting suitable morphologies. We selected terrain features that were invariant to the morphology of the robot – changing the length of the robot's legs did not affect the terrain measurements. We considered also adapting the control, but early testing showed that achieving the same invariance to morphology was much more challenging when varying the gait. More advanced methods for world modeling might be used to allow adaptation of control, which would give the robot even more flexibility than when only adapting morphology. This would most likely lead to the robot being able to adapt to an even more extensive range of terrains, supported by the fact that we saw significant differences in both morphology and control in all evolutionary experiments where they were both evolved.

Chapter 6

Conclusion

The main objective of the thesis was to develop methods and technology to enable adaptation of the physical body of a robot to new real-world environments. The papers in the thesis demonstrate evolutionary optimization of both control and morphology for offline adaptation to changing hardware conditions and walking surfaces, as well as a model-based continuous online adaptation of morphology in the real world.

The main contributions of the thesis can be summarised as follows:

- A mammal-inspired quadruped robot with self-adaptive leg lengths has been designed and built, and is available as an open-source hardware project.
- Evolutionary algorithms were shown to be able to exploit the dynamic morphology of a physical robot during co-optimization of control and body exclusively in the real world.
- A new gait controller concept with adjustable complexity was developed that allows the same controller to be used for both simulation and real-world experiments. It can also be adjusted to environmental requirements.
- A simple model-based online adaptation method was developed that can adapt the morphology of a robot efficiently during operation. This was demonstrated in the real world.
- The benefit of the robot's adaptive morphology was demonstrated both when experiencing changing hardware conditions and walking surfaces in the lab, as well as when walking on different terrains in the real world.

Evolutionary algorithms can successfully utilize the adaptive morphology of a mammal-inspired quadruped robot through experiments exclusively in the real world. The demonstrated feasibility and benefit of adapting morphology is important, especially when optimal leg shapes are unknown. This finding could extend to the field of robotics in general, where optimization is often pursued without considering different morphologies. The results of the thesis show promise for an embodied approach to solving challenging robotics problems in the real world.

Bibliography

[1] Ahmad, H., Nakata, Y., Nakamura, Y., and Ishiguro, H. "PedestriANS: a bipedal robot with adaptive morphology". In: *Adaptive Behavior* (2020).

[2] Auerbach, J. et al. "Robogen: Robot generation through artificial evolution". In: *Artificial Life Conference Proceedings 14*. MIT Press. 2014, pp. 136–137.

[3] Baines, R., Freeman, S., Fish, F., and Kramer, R. "Variable stiffness morphing limb for amphibious legged robots inspired by chelonian environmental adaptations". In: *Bioinspiration & Biomimetics* (2020).

[4] Bledt, G., Powell, M. J., Katz, B., Di Carlo, J., Wensing, P. M., and Kim, S. "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 2245–2252.

[5] Boeing, A. and Braunl, T. "Evolving Splines: An alternative locomotion controller for a bipedal robot". In: *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002*. Vol. 2. IEEE. 2002, pp. 798–802.

[6] Bongard, J. "Evolutionary robotics". In: *Communications of the ACM* vol. 56, no. 8 (2013), pp. 74–83.

[7] Bongard, J. "Morphological change in machines accelerates the evolution of robust behavior". In: *Proceedings of the National Academy of Sciences* vol. 108, no. 4 (2011), pp. 1234–1239.

[8] Bongard, J., Zykov, V., and Lipson, H. "Resilient machines through continuous self-modeling". In: *Science* vol. 314, no. 5802 (2006), pp. 1118–1121.

[9] Buzdalova, A. and Buzdalov, M. "Increasing efficiency of evolutionary algorithms by choosing between auxiliary fitness functions with reinforcement learning". In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 1. IEEE. 2012, pp. 150–155.

[10] Capi, G. and Doya, K. "Evolution of recurrent neural controllers using an extended parallel genetic algorithm". In: *Robotics and Autonomous Systems* vol. 52, no. 2-3 (2005), pp. 148–159.

[11] Chernova, S. and Veloso, M. "An evolutionary approach to gait learning for four-legged robots". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. Vol. 3. IEEE. 2004, pp. 2562–2567.

[12] Christensen, A. L., O'Grady, R., and Dorigo, M. "Morphology control in a multirobot system". In: *IEEE Robotics Automation Magazine* vol. 14, no. 4 (2007), pp. 18–25.

[13] Christensen, D. J., Campbell, J., and Stoy, K. "Anatomy-based organization of morphology and control in self-reconfigurable modular robots". In: *Neural Computing and Applications* vol. 19, no. 6 (2010), pp. 787–805.

[14] Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. "Evolving coordinated quadruped gaits with the HyperNEAT generative encoding". In: *2009 IEEE congress on evolutionary computation*. IEEE. 2009, pp. 2764–2771.

[15] Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer, 2007.

[16] Corucci, F., Calisti, M., Hauser, H., and Laschi, C. "Novelty-based evolutionary design of morphing underwater robots". In: *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*. 2015, pp. 145–152.

[17] Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. "Robots that can adapt like animals". In: *Nature* vol. 521, no. 7553 (2015), pp. 503–507.

[18] De Santos, P. G., Garcia, E., and Estremera, J. *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer Science & Business Media, 2007.

[19] Deb, K. and Jain, H. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints". In: *IEEE transactions on evolutionary computation* vol. 18, no. 4 (2013), pp. 577–601.

[20] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE transactions on evolutionary computation* vol. 6, no. 2 (2002), pp. 182–197.

[21] Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. G. "Evolutionary robotics: what, why, and where to". In: *Frontiers in Robotics and AI* vol. 2 (2015), p. 4.

[22] Doncieux, S., Mouret, J.-B., Bredeche, N., and Padois, V. "Evolutionary robotics: Exploring new horizons". In: *New horizons in evolutionary robotics*. Springer, 2011, pp. 3–25.

[23] Dubey, S., Prateek, M., and Saxena, M. "Robot locomotion–a review". In: *International Journal of Applied Engineering Research* vol. 10, no. 3 (2015), pp. 7357–69.

[24] Eiben, A. E. "Grand challenges for evolutionary robotics". In: *Frontiers in Robotics and AI* vol. 1 (2014), p. 4.

[25] Eiben, A. E. and Smith, J. E. *Introduction to evolutionary computing*. Springer, 2015.

[26] Gong, D., Yan, J., and Zuo, G. "A review of gait optimization based on evolutionary computation". In: *Applied Computational Intelligence and Soft Computing* vol. 2010 (2010).

[27] Graf, C., Härtl, A., Röfer, T., and Laue, T. "A robust closed-loop gait for the standard platform league humanoid". In: *Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the*. 2009, pp. 30–37.

[28] Guan, Y., Jiang, L., Zhangy, X., Zhang, H., and Zhou, X. "Development of novel robots with modular methodology". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 2385–2390.

[29] Heijnen, H., Howard, D., and Kottege, N. "A testbed that evolves hexapod controllers in hardware". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1065–1071.

[30] Hoepflinger, M. A., Remy, C. D., Hutter, M., Spinello, L., and Siegwart, R. "Haptic terrain classification for legged robots". In: *2010 IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 2828–2833.

[31] Hornby, G. S., Lipson, H., and Pollack, J. B. "Generative representations for the automated design of modular physical robots". In: *IEEE transactions on Robotics and Automation* vol. 19, no. 4 (2003), pp. 703–719.

[32] Hornby, G. S., Takamura, S., Yamamoto, T., and Fujita, M. "Autonomous evolution of dynamic gaits with two quadruped robots". In: *IEEE transactions on Robotics* vol. 21, no. 3 (2005), pp. 402–410.

[33] Howard, A. and Seraji, H. "Vision-based terrain characterization and traversability assessment". In: *Journal of Robotic Systems* vol. 18, no. 10 (2001), pp. 577–587.

[34] Hutter, M. et al. "Anymal-a highly mobile and dynamic quadrupedal robot". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 38–44.

[35] Ijspeert, A. J. "Central pattern generators for locomotion control in animals and robots: a review". In: *Neural networks* vol. 21, no. 4 (2008), pp. 642–653.

[36] Jakobi, N., Husbands, P., and Harvey, I. "Noise and the reality gap: The use of simulation in evolutionary robotics". In: *European Conference on Artificial Life*. Springer. 1995, pp. 704–720.

[37] Kawatsuma, S., Fukushima, M., and Okada, T. "Emergency response by robots to Fukushima-Daiichi accident: summary and lessons learned". In: *Industrial Robot: An International Journal* (2012).

[38] Kitano, S., Hirose, S., Endo, G., and Fukushima, E. F. "Development of lightweight sprawling-type quadruped robot titan-xiii and its dynamic walking". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 6025–6030.

[39] Knowles, J. D. and Corne, D. W. "Approximating the nondominated front using the Pareto archived evolution strategy". In: *Evolutionary computation* vol. 8, no. 2 (2000), pp. 149–172.

[40] Koos, S., Cully, A., and Mouret, J.-B. "Fast damage recovery in robotics with the t-resilience algorithm". In: *The International Journal of Robotics Research* vol. 32, no. 14 (2013), pp. 1700–1723.

[41] Koos, S., Mouret, J.-B., and Doncieux, S. "The transferability approach: Crossing the reality gap in evolutionary robotics". In: *IEEE Transactions on Evolutionary Computation* vol. 17, no. 1 (2012), pp. 122–145.

[42] Krasnogor, N. and Smith, J. "A tutorial for competent memetic algorithms: model, taxonomy, and design issues". In: *IEEE Transactions on Evolutionary Computation* vol. 9, no. 5 (2005), pp. 474–488.

[43] Krebs, A., Pradalier, C., and Siegwart, R. "Comparison of boosting based terrain classification using proprioceptive and exteroceptive data". In: *Experimental Robotics*. Springer. 2009, pp. 93–102.

[44] Kriegman, S., Walker, S., Shah, D., Levin, M., Kramer-Bottiglio, R., and Bongard, J. "Automated shapeshifting for function recovery in damaged robots". In: *Robotics: Science and Systems* (2019).

[45] Larsen, J. C. and Stoy, K. "LocoKit-A Construction Kit for Exploration of Morphology of Legged Robots". In: *in Proceedings of The International Symposium on Adaptive Motion in Animals and Machines*. Vol. 2011. 2011.

[46] Laschi, C., Mazzolai, B., and Cianchetti, M. "Soft robotics: Technologies and systems pushing the boundaries of robot abilities". In: *Sci. Robot* vol. 1, no. 1 (2016), eaah3690.

[47] Liang, X., Xu, M., Xu, L., Liu, P., Ren, X., Kong, Z., Yang, J., and Zhang, S. "The amphihex: A novel amphibious robot with transformable leg-flipper composite propulsion mechanism". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 3667–3672.

[48] Lim, S. H. and Teo, J. "Design, Optimization and Fabrication of a Climbing Six Articulated-Wheeled Robot Using Artificial Evolution and 3D Printing". In: *Journal of Advances in Mathematics and Computer Science* (2015), pp. 1–21.

[49] Lipson, H. and Pollack, J. B. "Automatic design and manufacture of robotic lifeforms". In: *Nature* vol. 406, no. 6799 (2000), pp. 974–978.

[50] Macinnes, I. and Di Paolo, E. “Crawling out of the simulation: Evolving real robot morphologies using cheap reusable modules”. In: *In Artificial Life IX: Proc. Ninth Intl. Conf. on the Simulation and Synthesis of Life*. 2004, p. 94.

[51] Martin, C. P., Glette, K., Nygaard, T. F., and Torresen, J. “Understanding Musical Predictions With an Embodied Interface for Musical Machine Learning”. In: *Frontiers in Artificial Intelligence* vol. 3 (2020). DOI: 10.3389/frai.2020.00006.

[52] Moravec, H. *Mind children: The future of robot and human intelligence.* Harvard University Press, 1988.

[53] Moreno, R., Veenstra, F., Silvera, D., Franco, J., Gracia, O., Cordoba, E., Gomez, J., and Faina, A. “Automated reconfiguration of modular robots using robot manipulators”. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2018, pp. 884–891.

[54] Mostafa, K., Tsai, C.-S., and Her, I. “Alternative gaits for multiped robots with leg failures to retain maneuverability”. In: *International Journal of Advanced Robotic Systems* vol. 7, no. 4 (2010), p. 34.

[55] Mouret, J.-B. and Chatzilygeroudis, K. “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.

[56] Nordmoen, J., Nygaard, T., Samuelsen, E., and Glette, K. “On Restricting Real-Valued Genotypes in Evolutionary Algorithms”. In: *arXiv preprint arXiv:2005.09380* (2020).

[57] Nordmoen, J., Nygaard, T. F., Ellefsen, K. O., and Glette, K. “Evolved Embodied Phase Coordination Enables Robust Quadruped Robot Locomotion”. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO ’19. Association for Computing Machinery, 2019, pp. 133–141. DOI: 10.1145/3321707.3321762.

[58] Nordmoen, J., Nygaard, T. F., Ellefsen, K. O., and Glette, K. “Evolved embodied phase coordination enables robust quadruped robot locomotion”. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2019, pp. 133–141. DOI: 10.1145/3321707.3321762.

[59] Nygaard, T. F., Glette, K., Martin, C., Torresen, J., and Howard, D. “A Morphologically Adaptive Quadruped Robot in the Wild”. In: *preprint* (2020).

[60] Nygaard, T. F., Howard, D., and Glette, K. “Real World Morphological Evolution is Feasible”. In: *2020 GECCO workshop on Evolution of Robots for the Real World* (2020).

[61] Nygaard, T. F., Martin, C., Howard, D., Torresen, J., and Glette, K. “Environmental Adaptation of Robot Morphology and Control through Real-world Evolution”. In: *preprint* (2020).

[62] Nygaard, T. F., Martin, C. P., Samuelsen, E., Torresen, J., and Glette, K. “Real-world Evolution Adapts Robot Morphology and Control to Hardware Limitations”. In: *2018 Genetic and Evolutionary Computation Conference (GECCO18)*. Kyoto, Japan: ACM, 2018, pp. 125–132. DOI: 10.1145/3205455.3205567.

[63] Nygaard, T. F., Martin, C. P., Torresen, J., and Glette, K. “Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds”. In: *Applications of Evolutionary Computation*. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-16692-2_41.

[64] Nygaard, T. F., Martin, C. P., Torresen, J., and Glette, K. “Exploring Mechanically Self-Reconfiguring Robots for Autonomous Design”. In: *2018 ICRA Workshop on Autonomous Robot Design* (2018).

[65] Nygaard, T. F., Martin, C. P., Torresen, J., and Glette, K. “Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing”. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. 2019. DOI: 10.1109/ICRA.2019.8793663.

[66] Nygaard, T. F., Nordmoen, J., Ellefsen, K. O., Martin, C. P., Tørresen, J., and Glette, K. “Experiences from Real-World Evolution with DyRET: Dynamic Robot for Embodied Testing”. In: *Symposium of the Norwegian AI Society*. Springer. 2019, pp. 58–68. DOI: 10.1007/978-3-030-35664-4_6.

[67] Nygaard, T. F., Nordmoen, J., Martin, C. P., and Glette, K. “Lessons Learned from Real-World Experiments with DyRET: the Dynamic Robot for Embodied Testing”. In: *2019 ICRA Workshop on Learning Legged Locomotion* (2019).

[68] Nygaard, T. F., Samuelsen, E., and Glette, K. “Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach”. In: *European Conference on the Applications of Evolutionary Computation*. Springer. 2017, pp. 825–836. DOI: 10.1007/978-3-319-55849-3_53.

[69] Nygaard, T. F., Torresen, J., and Glette, K. “Multi-objective evolution of fast and stable gaits on a physical quadruped robotic platform”. In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2016, pp. 1–8. DOI: 10.1109/SSCI.2016.7850167.

[70] Ocenasek, J., Kern, S., Hansen, N., and Koumoutsakos, P. “A mixed bayesian optimization algorithm with variance adaptation”. In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2004, pp. 352–361.

[71] Ojeda, L., Borenstein, J., Witus, G., and Karlsen, R. “Terrain characterization and classification with a mobile robot”. In: *Journal of Field Robotics* vol. 23, no. 2 (2006), pp. 103–122.

[72] Paik, J. K., Byoungkwon, A., Rus, D., and Wood, R. J. "Robotic origamis: Self-morphing modular robot". In: *Proceedings of the 2nd International Conference on Morphological Computation*. 2012.

[73] Parker, G. B. and Georgescu, R. A. "Using cyclic genetic algorithms to evolve multi-loop control programs". In: *IEEE International Conference Mechatronics and Automation, 2005*. Vol. 1. IEEE. 2005, pp. 113–118.

[74] Picardi, G., Hauser, H., Laschi, C., and Calisti, M. "Morphologically induced stability on an underwater legged robot with a deformable body". In: *The International Journal of Robotics Research* (2019).

[75] Pugh, J. K., Soros, L. B., and Stanley, K. O. "Quality diversity: A new frontier for evolutionary computation". In: *Frontiers in Robotics and AI* vol. 3 (2016), p. 40.

[76] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[77] Rahmat, N. A. and Musirin, I. "Differential evolution ant colony optimization (DEACO) technique in solving economic load dispatch problem". In: *2012 IEEE International Power Engineering and Optimization Conference Melaka, Malaysia*. IEEE. 2012, pp. 263–268.

[78] Rhodes, T., Gotberg, C., and Vikas, V. "Compact Shape Morphing Tensegrity Robots Capable of Locomotion". In: *Frontiers in Robotics and AI* vol. 6 (2019), p. 111.

[79] Ruud, E.-L., Samuelsen, E., and Glette, K. "Memetic robot control evolution and adaption to reality". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2016, pp. 1–7.

[80] Samuelsen, E. and Glette, K. "Real-world reproduction of evolved robot morphologies: Automated categorization and evaluation". In: *European Conference on the Applications of Evolutionary Computation*. Springer. 2015, pp. 771–782.

[81] Sayyad, A., Seth, B., and Seshu, P. "Single-legged hopping robotics research—A review". In: *Robotica* vol. 25, no. 5 (2007), pp. 587–613.

[82] Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. "Design of HyQ–a hydraulically and electrically actuated quadruped robot". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* vol. 225, no. 6 (2011), pp. 831–849.

[83] Shen, H., Yosinski, J., Kormushev, P., Caldwell, D. G., and Lipson, H. "Learning fast quadruped robot gaits with the RL power spline parameterization". In: *Cybernetics and Information Technologies* vol. 12, no. 3 (2012), pp. 66–75.

[84] Shen, S.-Y., Li, C.-H., Cheng, C.-C., Lu, J.-C., Wang, S.-F., and Lin, P.-C. "Design of a leg-wheel hybrid mobile platform". In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 4682–4687.

[85] Sims, K. "Evolving 3D morphology and behavior by competition". In: *Artificial life* vol. 1, no. 4 (1994), pp. 353–372.

[86] Smith, B. G., Saaj, C. M., and Allouis, E. "Evolving legged robots using biologically inspired optimization strategies". In: *2010 IEEE International Conference on Robotics and Biomimetics*. IEEE. 2010, pp. 1335–1340.

[87] Sun, Y., Chen, X., Yan, T., and Jia, W. "Modules design of a reconfigurable multi-legged walking robot". In: *2006 IEEE International Conference on Robotics and Biomimetics*. IEEE. 2006, pp. 1444–1449.

[88] Todd, D. J. *Walking machines: an introduction to legged robots*. Springer Science & Business Media, 2013.

[89] Toffolo, A. and Benini, E. "Genetic diversity as an objective in multi-objective evolutionary algorithms". In: *Evolutionary computation* vol. 11, no. 2 (2003), pp. 151–167.

[90] Ursem, R. K. "Diversity-guided evolutionary algorithms". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2002, pp. 462–471.

[91] Usevitch, N. S., Hammond, Z. M., Schwager, M., Okamura, A. M., Hawkes, E. W., and Follmer, S. "An untethered isoperimetric soft robot". In: *Science Robotics* vol. 5, no. 40 (2020).

[92] Van, R., Sugar, T., Vanderborght, B., Hollander, K., and Lefeber, D. "Compliant actuator designs. Review of actuators with passive adjustable compliance/controllable stiffness for robotic applications". In: *IEEE Robotics Automation Magazine* vol. 16, no. 3 (2009), pp. 81–94.

[93] Vanderborght, B. et al. "Variable impedance actuators: A review". In: *Robotics and autonomous systems* vol. 61, no. 12 (2013), pp. 1601–1614.

[94] Vujovic, V., Rosendo, A., Brodbeck, L., and Iida, F. "Evolutionary developmental robotics: Improving morphology and control of physical robots". In: *Artificial Life* (2017).

[95] Weiss, C., Fechner, N., Stark, M., and Zell, A. "Comparison of Different Approaches to Vibration-based Terrain Classification." In: *EMCR*. 2007.

[96] Wieber, P.-B., Tedrake, R., and Kuindersma, S. "Modeling and control of legged robots". In: *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.

[97] Wilson, A. D. and Golonka, S. "Embodied cognition is not what you think it is". In: *Frontiers in psychology* vol. 4 (2013), p. 58.

[98] Wolff, K., Sandberg, D., and Wahde, M. "Evolutionary optimization of a bipedal gait in a physical robot". In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 440–445.

[99] Wright, J. and Jordanov, I. "Intelligent approaches in locomotion-a review". In: *Journal of Intelligent & Robotic Systems* vol. 80, no. 2 (2015), pp. 255–277.

[100] Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J. C., and Lipson, H. "Evolving robot gaits in hardware: the HyperNEAT generative encoding vs. parameter optimization." In: *ECAL*. 2011, pp. 890–897.

[101] Zagal, J. C. and Ruiz-Del-Solar, J. "Combining simulation and reality in evolutionary robotics". In: *Journal of Intelligent and Robotic Systems* vol. 50, no. 1 (2007), pp. 19–39.

[102] Zhang, T., Zhang, W., and Gupta, M. M. "Resilient Robots: Concept, Review, and Future Directions". In: *Robotics* vol. 6, no. 4 (2017), p. 22.

[103] Zitzler, E., Laumanns, M., and Thiele, L. "SPEA2: Improving the strength Pareto evolutionary algorithm". In: *TIK-report* vol. 103 (2001).

Papers

Paper I

Multi-objective Evolution of Fast and Stable Gaits on a Physical Quadruped Robotic Platform

Tønnes F. Nygaard, Jim Torresen, Kyrre Glette

2016 IEEE Symposium Series on Computational Intelligence
ISBN: 978-1-5090-4240-1, DOI: 10.1109/SSCI.2016.7850167

Multi-objective Evolution of Fast and Stable Gaits on a Physical Quadruped Robotic Platform

Tønnes F. Nygaard, Jim Torresen, Kyrre Glette
Department of Informatics
University of Oslo
P.O. Box 1080 Blindern, 0316 Oslo, Norway
Email: tonnesfn,jimtoer,kyrrehg@ifi.uio.no

Abstract – The field of evolutionary robotics shows great promise, but is held back by the lack of results applicable to real world problems or other research fields. The reality gap effects present when moving from virtual to real robots makes evolution based on simulation inefficient for continuous adaption to changing morphology or environments. Evolution on the physical robot does not share these challenges, but each experiment in hardware is limited by the high time requirement of each evaluation. In this paper we suggest using a high level controller with multi-objective optimization of speed and stability to achieve a range of robust gaits for a quadruped robot that does not require excessive tests on the real robot. Using multi-objective evolutionary optimization on the physical robot, we achieved a Pareto front with high performing and robust individuals showing different trade-offs between speed and stability. Single objective optimization of either speed or stability did not yield individuals with a trade-off between the two objective functions. The results show that multi-objective evolutionary optimization on the physical robot is not only feasible, but preferable over using single-objective optimization, given a high level gait controller.

1 Introduction and related work

The field of evolutionary robotics (ER) uses evolutionary computation to automatically optimize robot controllers and morphologies [1]. This process involves iteratively generating new candidate solutions and evaluating them in simulation or on the physical robot. ER methods have been used both as a design tool and for making continuous adaptions to changing situations or environments.

Many of the techniques used in traditional robotics for design of walking gaits or robot morphologies require a team of experienced engineers and excessive resources and time for trial and error. Automatic parameter tuning can help reduce the number of iterations, and use of evolutionary aided design can give an engineer new ways to analyze the problem and give suggestions for new design features [2]. Use of evolutionary algorithms in robotics can serve as a tool to save time during development, but examples have also been seen where evolved solutions outperformed hand designed solutions [3].

Changes in the robot [4] or its environment [5] can greatly alter the quality of a given controller or morphology. Walking on asphalt and through soft sand will most likely demand different walking gaits [6], and using techniques from evolutionary robotics to quickly and efficiently evolve a new solution on the physical robot that better handles the new surface could provide adaptability to a wide range of different environments and situations. This will become even more important in the future as robots are used for more advanced tasks and in more complex environments.

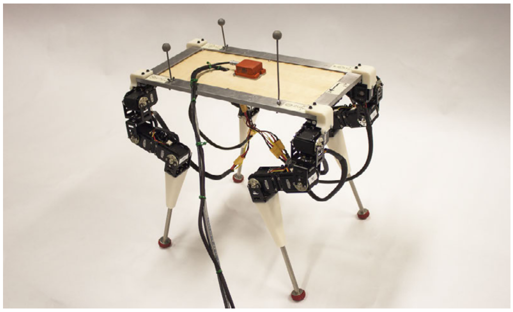


Figure 1: The robot used for the experiments.

Each evaluation on the physical robot requires a few seconds [7], to several minutes [3] of evaluation time, and much of the previous research involves simple, low level controllers that require a large number of evaluations to yield good results [8, 9]. The long duration of the experiments is one of the main reasons that make evolution on physical robots a difficult task, in addition to high mechanical wear on the robots used and excessive inaccuracies and measurement noise, when compared to a typical simulation. Most of the previous research in evolution of legged robot controllers on the physical robot is still fundamental research with focus on the main theoretical principles, and proof of concept experiments, rather than attempting to solve complex real world problems. This has resulted in many mechanically simple and fairly limited robots being used in current evolutionary experiments [4, 7, 10], though we are starting to see more capable robots emerging from the traditional robotics field [11, 12] which might serve as evolutionary platforms for real world applications in the future.

Each run of an algorithm may involve thousands of evaluations, and tests on physical robots are therefore often difficult or impossible, given the long duration of each evaluation. Simulation is used extensively to enable more efficient experimentation, but suffer from reality gap effects which in many cases make the solutions found less optimal in the real world [13]. There are techniques to lessen the difference between simulations and the physical world, including use of added noise in parameters [13] or simulated environment [14]. Research in combining the quick but inaccurate simulations with slow but accurate real world evaluations has been performed [4, 10, 15], but no standardized solution has been adopted into wide use.

Most of the previous work in evolution on physical robots optimizes a single objective, due to the long duration of evaluations. These single-objective (SO) evolutionary runs produce reasonable gaits for simple robots, especially where the possibility of tipping or falling is minimal. More advanced robots, however, often require more than one objective to achieve feasible gaits [16], though they require a substantially higher number of evaluations than SO optimization [17]. Some experiments have been done using multi-objective optimization (MO) in simulation, with transferal to a real legged robot [18], but these suffer from reality gap effects and the inability to be used continuously with on-line improvement on the actual robot. Examples of combining several objective functions into a single fitness function using weighted sum fitness have also been used successfully [19]. This does, however, only result in individuals with the chosen trade-off between objective functions, and does not return a Pareto front from which solutions with different trade-offs can be selected after evolution.

Sharing of ideas and principles between research fields is important to speed up innovation and generate interest and motivation. For evolutionary robotics to be relevant to other fields, more *robust* and *general* robots with the ability to serve as tools by other researchers need to be developed. Our goal is to show that Pareto-based multi-objective optimization of gaits on a four legged robot produces more versatile and robust solutions than running single-objective optimization, and that it is possible to perform this on the physical robot, thereby avoiding reality gap effects present in individuals evolved in simulation.

In this paper, we use a four legged robot with relatively powerful servos and a high level control system that uses inverse kinematics from classical robotics. We run multi-objective NSGA-II optimization of gaits with speed and stability as objective functions, to achieve a robust gait with a range of different speeds and stabilities for various applications. Single-objective runs optimizing each objective individually are also performed to demonstrate the differences between results from the multi-objective and single-objective cases. We also select the top performing gaits from each SO run, and a selection of gaits from the Pareto front resulting from the MO optimization, and compare and verify the performance by re-running the individual gaits.

We have not seen any previous work doing multi-objective optimization on a physical four legged robot resulting in a Pareto front with stable and robust gaits. Our use of a high level controller limits the number of invalid solutions, while still allowing the freedom for a range of different gaits.

The implementation is shown in section 2, describing the robot, control system, evolutionary system, and physical test setup. Section 3 describes the experiments performed and the observed results, followed by discussion in section 4 and conclusion and avenues for future work in section 5.

2 Robot and evolutionary setup

2.1 Robot

All experiments were performed using a custom robotic platform currently under development at the University of Oslo, which can be seen in Fig. 1. The top frame is made of aluminum, and measures 420mm * 220mm, with a plywood center. The four legs are about 45cm long, connected by aluminum brackets

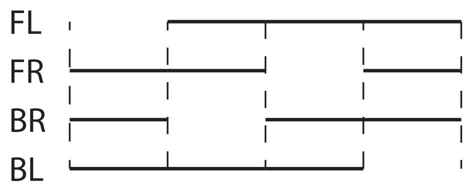


Figure 2: The gait sequence of the robot. Solid lines mark ground contact, and leg positions are given according to front (F), back (B), left (L) and right (R).

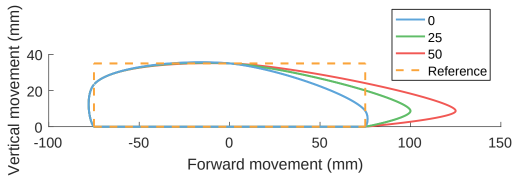


Figure 3: The resulting spline with *step_length* 150, *step_height* 35, and three different *step_smoothing* parameters. A reference rectangle with the chosen *step_height* and *step_length* is shown for comparison.

with a 3D printed ABS upper tibia, and are placed in a mammal configuration. They each feature three *Dynamixel MX-64* servos, with onboard PID controllers receiving the commanded angles over USB. An *Xsens MTI-30 attitude and heading reference system (AHRS)* is mounted in the middle of the body to measure linear acceleration, rotational velocity and magnetic fields, giving data on absolute orientation. Reflective markers are mounted on the top plate to allow for using motion capture equipment to record the position and orientation of the robot. The complete robot weights about 4.5kg, and operates tethered during all experiments.

2.2 Control system

A continuous, regular crawl gait [20] was chosen for its capacity of constant forward speed. The gait sequence can be seen in Fig. 2. The body moves steadily during the gait sequence, and each leg lifts and moves to the front incrementally to maximize ground contact time and stability. Constant movement can be advantageous when collecting sensor data of the environment, or using one of many mapping algorithms [21]. The path for each individual leg end is defined by a spline, and the centripetal catmull rom spline [22] was chosen for its interpolating nature and relative resistance to self-intersection.

The gait generator use parameter ranges defined in Table 1 and generates a number of control points for the spline, resulting in a continuous gait path for each leg. Three parameters are used for manipulating the control points. The parameter *step_length* controls the length of the ground contact line, while *step_height* determines the height of the step. The *step_smoothing* parameter regulates the angle of movement in the point where the leg end hits the ground, by stretching out the front of the spline. This was added to allow for a reduction of the impact forces from each step, by making contact with the ground in a more

Category	Name	Values
Spline shape	*step_length*	[0, 150mm]
	step_height	[0, 50mm]
	step_smoothing	[0, 50mm]
Gait timing	*gait_frequency*	[0, 1.5hz]
Balancing	*wag_phase*	[−0.2, 0.2]
	wag_amplitude_x	[0, 50mm]
	wag_amplitude_y	[0, 50mm]

Table 1: Parameters for the gait generation.

Algorithm	NSGA-II
Evaluation time	Maximum 50s
Parameters	Real: [0, 1]
Recombination	None
Mutation	Type: Gaussian
	Probability: 1.0
	Sigma: 1/6

Table 2: Parameters for the evolutionary setup

horizontal direction. Examples of a typical spline with different *step_smoothing* parameters can be seen in Fig. 3.

To increase the stability of the gait, a wag was added where the robot at all times leans to the opposite side of the currently lifted leg. This ensures a higher margin of stability and is required for a statically stable gait, due to the relatively high mass of the legs compared to the body. This wag movement has a phase offset to correct for differing control delays between the walking motion and the wag, and feature different amplitudes for lengthwise and sidewise movement. The gait has a *gait_frequency* parameter that, together with the *step_length*, forms the speed of the robot.

The control system is implemented in C++ using the software framework *Robot Operating System (ROS)* [23]. The leg end positions from the gait controller are sent through an inverse kinematics function to obtain the angles necessary to achieve the specified pose. The different functions of the robot controller are implemented as individual ROS nodes, and runs on a laptop connected to the servos and AHRS by cable.

2.3 Evolutionary setup

The software running the evolutionary algorithm uses Sferes2 [24], a C++ framework for evolutionary experiments. The NSGA-II algorithm was chosen for both SO and MO runs to ease comparison of results from the different runs. When it optimizes a single-objective, it reduces to a binary tournament-based evolutionary algorithm with truncating survivor selection.

Real values with a range from 0 to 1 were chosen to represent the genotype. These are then scaled to the values in table 1 when testing a candidate. Gaussian mutation is used on all genes with a sigma of 1/6, while no recombination is used, as seen in table 2.

Two objective functions are used in the experiments in this paper, *speed* and *stability*. The speed is calculated by using the duration of the gait and the Euclidean distance between the start and end position captured by the *motion capture equipment*, as seen in equation 1. An objective function for stability using only the gyro within the AHRS has been used in similar cases, but we observed in initial tests that gaits that were qualitatively perceived as very unstable received high gyro-based stability scores. A new objective function using both the *orientation* and measured *linear acceleration* from the *AHRS* sampled at 100hz was used instead, and provided a much closer match between perceived qualitative stability and measured quantitative stability fitness. The full stability objective function, seen in equation 4, is a sum of the linear acceleration function and the orientation function, seen in equations 2 and 3, where *acc* is a single sample from the accelerometer, *i* is the sample index, and *j* is the axis of the sample. *Roll* and *pitch* are orientation angles obtained directly from the *AHRS*. The *scalingFactor* was chosen to provide a balance between the two functions by having acceleration and orientation affect the fitness value equally, and was in these experiments set to 50. The stability objective function is negated to allow for maximization of both objective functions.

$$fitness_{speed} = \frac{dist(position_{start}, position_{end})}{time_{end} - time_{start}} \tag{1}$$

$$F_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(acc_{i,j} - \overline{acc_j})^2} \tag{2}$$

$$G = \sqrt{\frac{1}{N}\sum_{i=1}^{N} roll_i^2} + \sqrt{\frac{1}{N}\sum_{i=1}^{N} pitch_i^2} \tag{3}$$

$$fitness_{stability} = -\left(\frac{F_x + F_y + F_z}{scaling\_factor} + G\right) \tag{4}$$

2.4 Physical test setup

The goal of the physical test setup is to maximize the quality of measurements, while minimizing down time and requirements for human intervention. Motion capture equipment is used to provide a precise and accurate reading of position for estimation of speed. The time for each measurement is chosen to provide a good balance between many inaccurate measurements, and few but accurate evaluations, given a set time budget. Each evaluation is obtained by walking one meter forward, and then using the same gait back to the start position, before averaging the fitness values achieved for both directions. Walking in each of the two directions is restricted by a timeout of 10 seconds, to limit the time spent on evaluating slower individuals.

Both the robot and control system are designed to ensure repeatability for gaits by keeping the distance moved between each evaluation minimal. This is achieved by having the robot sequentially lift and reposition the legs to the start pose of the new gait after each evaluation. Two walking sequences of 10 seconds, in addition to repositioning of legs before and after the gait, results in a maximum of 50 seconds used for each evaluation. Some human intervention is required when the drift between evaluations has become too large, however, to move the robot back to the center of the test area. This has been observed to be once every three to ten minutes, depending on the objective and stage of evolution. If the robot falls or finishes evaluation without the body being parallel to the floor, the program pauses and waits for human intervention before continuing, typically happening about every 30 minutes.

3 Experiments and Results

3.1 Experiments

Evolutionary runs were performed with three different configurations: an SO run optimizing speed, an SO run optimizing stability, and an MO run optimizing *both* speed and stability. Parameters for the different runs are given in table 3. To make a direct comparison between results of the Pareto front and the results from the single-objective runs possible, one of each single-objective run was

Objectives	Population	Generations	Max time per run
Speed	8	16	1h 47m
Stability	8	16	1h 47m
Speed, Stability	32	8	3h 34m

Table 3: Parameters for the different evolutionary runs.

compared to the multi-objective run. This ensures comparisons of results from the same number of evaluations, since the MO run has twice the number of evaluations as an SO run.

A number of well performing individuals from the final population of the evolutionary runs are selected for re-evaluation and detailed analysis. This is important both to confirm the validity of the measured fitness, and to generate additional information on the performance of each individual for analysis and graphing.

3.2 Results

This section first shows the results of the two different SO runs, before presenting the results from the MO run. The results of the multiple re-evaluations of the top performers are presented last.

The SO optimization of speed resulted in the fitnesses seen in Fig. 4. The figure shows a relatively high initial speed in the randomized initial population, and we see a steady rise in speed through all generations. Stability decreases in the majority of the run, but has a slight increase in the last individual found. Fig. 5 shows an initial maximization of *step_length*, and a *gait_frequency* at approximately the middle of the allowed values. The *gait_frequency* rises steadily through the generations, and we see a slight decrease in *step_length* towards the end.

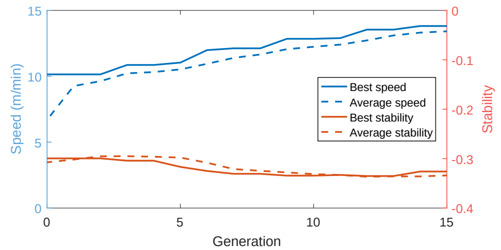
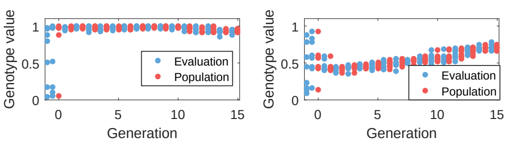


Figure 4: Fitness results from speed optimization.



(a) Evaluation and selection of *step_length* values.

(b) Evaluation and selection of *gait_frequency* values.

Figure 5: Parameters in the SO optimization of speed. The changing population is seen, as well as the evaluated parameters between each generation.

The fitness from the SO optimization of stability can be seen in Fig. 6. This figure shows convergence after only 4 generations, though with a slight increase in fitness in the last three generations.

Fig. 7 shows which parameters are tested throughout the run, and we can see that *step_length* is centered around 20% throughout the run, while the *gait_frequency* is quickly minimized.

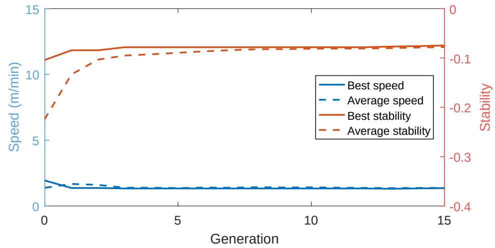
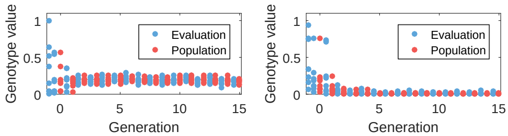


Figure 6: Fitness results from stability optimization.



(a) Evaluation and selection of *step_length* values.

(b) Evaluation and selection of *gait_frequency* values.

Figure 7: Parameters in the SO optimization of stability. The changing population is seen, as well as the evaluated parameters between each generation.

Fig. 8 show the results from the MO run optimizing both speed and stability. The Pareto optimal solutions follow a slightly curved shape with three large holes in the front, with a few barely dominated solutions shortening in a couple of the gaps. We can also see the results from both of the SO runs in the same figure. We see that both runs outperform the solutions found in the Pareto front by a relatively small amount, but are concentrated along the two extremes, yielding no viable solutions with any trade-off between the different objectives.

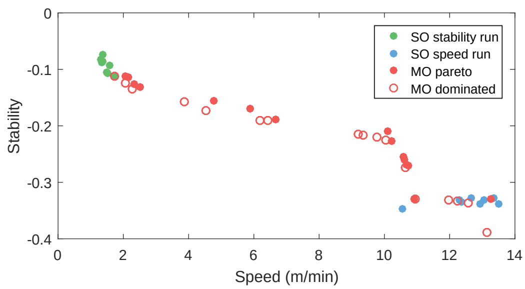
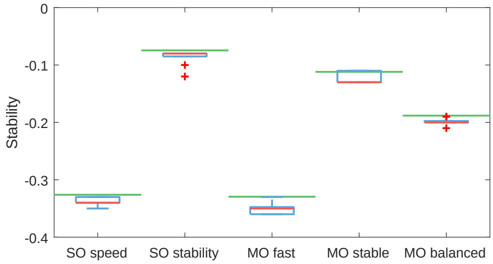
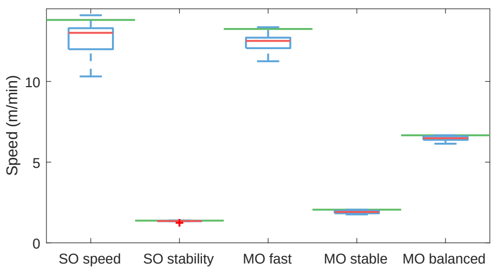


Figure 8: Pareto front from the MO run, along with individuals from the SO runs of speed and stability.

Fig. 9 shows the re-evaluated solutions from the evolutionary experiments, where the fitness from the selected individuals are verified by running each of the gaits an additional 10 times. The top performing individual from each SO run was tested, as well as one individual from each extreme, and one individual from the middle of the Pareto front. We see that the measurement noise



(a) Box plot of stability from re-evaluations.



(b) Box plot of speed from re-evaluations.

Figure 9: Box plot showing the 10 new fitness measurements on each of the selected individuals, with original fitness values from the evolutionary run in green.

is fairly low for both objective functions, though slightly higher for speed. We also see no large discrepancies between the original fitness measurements during the evolutionary run in green, and the distribution of measured fitnesses from the re-evaluations.

Fig. 10 shows one of the resulting gait splines from a selection of individuals from the runs, in addition to raw measurements of the two fitnesses, here given by distance covered and stability. We see from the figure that the individuals from the extremes of the Pareto front resemble the individuals found in the SO runs, and that the individual with a trade-off between stability and speed more closely resembles the high stability individuals on the shape of the spline and the stability achieved, while the distance covered appear more similar to the speed optimized individuals.

4 Discussion

We made the following observations from the results:

- The SO optimization runs slightly outperformed the extremes of the Pareto front from the MO run, as seen in Fig. 8, but none of the results from any of the SO runs are directly useable in most applications. A high speed is achieved using SO optimization, as seen in Fig. 4, but it will easily tip over, and the performance will most likely suffer with a slightly different weight distribution on the robot or a different ground friction, as the stability objective function is low. Fig. 6 shows a high stability from the SO stability optimization, but the speed is so low that the applications are very limited. The MO run produces slightly worse performing individuals than the two SO runs in the extremes of the front, but provides a range of choices with different trade-offs between stability and speed throughout the front. More runs or a larger population would most likely make the gaps smaller. This shows that SO optimization of either stability or speed is ineffective, while MO evolution produced a range of suitable gaits with different trade-offs between speed and stability, with a much higher relevance to real world problems.

- The use of a high level control system severely reduces the number of infeasible gaits tested on the robot, although it also limits the diversity of different types of gaits. Many lower level controllers have been successfully used in both simulation and single-objective evolution on a physical robot, but require a high number of evaluations that makes it infeasible to do multi-objective evolutionary optimization on real robots alone. We see from the fitness graphs in Fig. 10 that several of the randomized solutions in the initial populations do relatively well, and that as few as 32 evaluations are enough to achieve a stable gait in its corresponding single-objective experiment. This shows that the control system used is a good choice when facing time consuming evaluations like we do when evolving on the physical robot, and the highest achieved crawl gait speed of about 23cm/s is considered very good for the small number of evaluations used.

- We can see from Fig. 10 that the most stable solution with stability in Fig. 10f also has one of the least constant speeds, seen in the distance walked in Fig. 10e. This might seem counter intuitive given that stability is dependent on low variation in linear acceleration, but this shows that a varying speed of the body is needed to achieve high stability by counteracting the relatively large mass of the legs. We also see that the individual featuring a trade-off between stability and speed has a spline that resembles the individuals from the high stability runs. This indicates that the major difference between the slow and stable individual, and the fast and stable individual is mainly found in timing and balancing, and not in the shape of the spline.

- Fig. 9 shows a relatively low variation in fitness measurements over the 10 re-evaluations of each individual. We also see that the original fitness measurements taken during the evolution correspond to the re-evaluated fitnesses tested a few days later. This shows a high degree of repeatability in the test setup, which requires low measurement noise, predictable gait generation, and precise control of the robot. High variation would give many of the same challenges seen when experiencing reality gap effects, but this has shown not to be the case with this robot and experimental setup.

- Optimizing lower level control systems by hand can be a challenging task. The parameters are often not connected directly to the physical robot, and it can be hard or impossible for an engineer to predict how changing certain parameters would affect the end result. All parameters of the high level control system we use, shown in table 1, are easy to visualize and understand. Not only does this make it easier for an engineer to design gaits using this controller, but it makes directly comparing hand designed gaits by an engineer to evolved or otherwise automatically

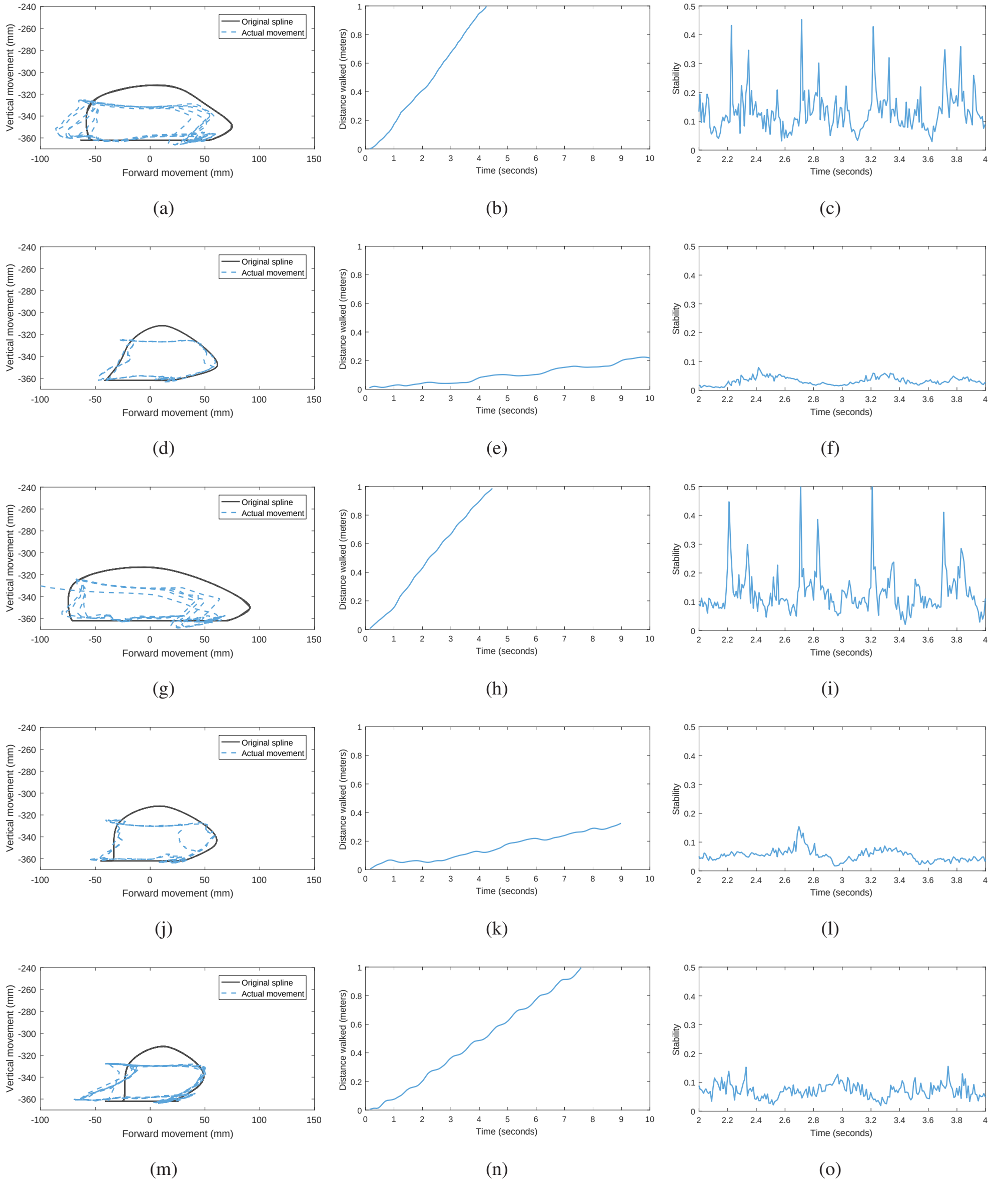


Figure 10: First column is the gait spline of the individual, second column is the distance from the starting position, and the third column is the stability of the gait. Subfigure (a) to (c) is the fastest individual from the SO optimization of speed, (d) to (f) is the most stable from the SO optimization of stability. The rest of the subfigures are individuals from the MO run optimizing both speed and stability, with (g) to (i) being the fastest individual, (j) to (l) being the most stable, and (m) to (p) being a trade-off between the two from the center of the Pareto front.

optimized gaits less ambiguous. Using evolution on a more intuitive controller also promotes more efficient use of the evolutionary results when doing evolutionary aided design, with easier analysis and better human understanding of the resulting parameters.

5 Conclusion and future work

In this paper we have investigated using both single and multi-objective evolutionary optimization on the physical robot to generate parameters for the high level controller producing a continuous, regular wave gait for a four legged robot. A physical test setup is used which provides robust fitness measurements with low noise and high repeatability between evaluations of the same gait parameters. We saw that the high level controller made it possible to achieve high performing individuals after a small number of evaluations, which makes multi-objective optimization a feasible method for gait generation on the physical robot. Evolved individuals from the SO runs performed well in regards to their goal, but lack a robust gait with real world applicability. Gaits from the MO run feature a range of different trade-offs between stability and speed, and therefore higher relevance to a range of applications.

It would be interesting to test some of the solutions with low stability and evaluate performance on surfaces with different friction, inclinations, and obstacles to see how well the stability measurement corresponds to robustness. Continuing the work on the high level controller to allow for a more diverse set of gaits, while still not generating a high degree of infeasible individuals, could yield solutions with an even lower number of required evaluations. Using SO optimization of a weighted sum fitness function of a combination of speed and stability might, in combination with other techniques for preserving diversity, yield similar results to multi-objective methods. The use of an archive scheme, previously used for instance to allow a robot to walk in all directions [25], might enable a single-objective algorithm to present a range of alternative solutions, comparable to the Pareto front of the MO run. Moving away from using motion capture and instead using the AHRS to measure the speed would decrease the complexity of the system and enable more labs to use the system. Working on reducing the evaluation time by accepting more noise in the fitness values would enable more and larger evolutionary experiments on the current controller, or enable more complex controllers to be used. Furthermore, adding the ability to do simulations on the system and incorporating that to reduce the number of unneeded evaluations on the physical robot might further reduce the need for lengthy hardware trials, though many challenges will arise from the reality gap effects.

Acknowledgment

This work is partially supported by The Research Council of Norway as a part of the Engineering Predictability with Embodied Cognition (EPEC) project, under grant agreement 240862.

References

[1] S. Doncieux, J.-B. Mouret, N. Bredeche, and V. Padois, "Evolutionary robotics: Exploring new horizons," in *New horizons in evolutionary robotics*. Springer, 2011, pp. 3–25.

[2] S. Hauert, J. C. Zufferey, and D. Floreano, "Reverse-engineering of artificially evolved controllers for swarms of robots," in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 55–61.

[3] G. S. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita, "Evolving robust gaits with aibo," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 3, 2000, pp. 3040–3045 vol.3.

[4] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, 2006.

[5] M. A. Gongora, B. N. Passow, and A. A. Hopgood, "Robustness analysis of evolutionary controller tuning using real systems," in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 606–613.

[6] W. Bosworth, J. Whitney, S. Kim, and N. Hogan, "Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3582–3589.

[7] S. Lee, J. Yosinski, K. Glette, H. Lipson, and J. Clune, "Evolving gaits for physical robots with the hyperneat generative encoding: The benefits of simulation," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2013, pp. 540–549.

[8] 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 Seventeenth International Symposium on Artificial Life and Robotics*, 2012, pp. 1–4.

[9] T. Gomi and K. Ide, "Evolution of gaits of a legged robot," in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, vol. 1. IEEE, 1998, pp. 159–164.

[10] 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, no. 1, pp. 122–145, Feb 2013.

[11] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *15th International Conference on Climbing and Walking Robot*, 2012.

[12] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011.

[13] 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.

[14] K. Glette, A. L. Johnsen, and E. Samuelsen, "Filling the reality gap: Using obstacles to promote robust gaits in evolutionary robotics," in *Proceedings of the 2014 IEEE International Conference on Evolvable Systems (ICES)*. IEEE, 2014, pp. 181–186.

[15] J. C. Zagal, J. Ruiz-del Solar, and P. Vallejos, "Back to reality: Crossing the reality gap in evolutionary robotics,"

in *IAV 2004 the 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal*, 2004.

[16] D. Gong, J. Yan, and G. Zuo, "A review of gait optimization based on evolutionary computation," *Applied Computational Intelligence and Soft Computing*, vol. 2010, 2010.

[17] V. Khare, X. Yao, and K. Deb, *Performance Scaling of Multi-objective Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 376–390.

[18] E. Samuelsen, K. Glette, and J. Torresen, "A hox gene inspired generative approach to evolving robot morphology," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 751–758.

[19] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 345–370, 2009.

[20] P. G. de Santos, E. Garcia, and J. Estremera, *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer Science & Business Media, 2007.

[21] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008.

[22] C. Yuksel, S. Schaefer, and J. Keyser, "On the parameterization of catmull-rom curves," in *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*. ACM, 2009, pp. 47–53.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[24] J.-B. Mouret and S. Doncieux, "SFERESv2: Evolvin' in the multi-core world," in *Proc. of Congress on Evolutionary Computation (CEC)*, 2010, pp. 4079–4086.

[25] A. Cully and J.-B. Mouret, "Evolving a behavioral repertoire for a walking robot," *Evolutionary computation*, vol. 24, no. 1, pp. 59–88, 2016.

Paper II

Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing

II

Tønnes F. Nygaard, Charles P. Martin, Jim Torresen, Kyrre Glette

2019 IEEE International Conference on Robotics and Automation
ISBN: 978-1-5386-6027-0, DOI: 10.1109/ICRA.2019.8793663

Self-Modifying Morphology Experiments with DyRET: Dynamic Robot for Embodied Testing

Tønnes F. Nygaard[1], Charles P. Martin[1], Jim Torresen[1,2] and Kyrre Glette[1,2]
[1] The Department of Informatics, University of Oslo
[2] RITMO, University of Oslo

Abstract – If robots are to become ubiquitous, they will need to be able to adapt to complex and dynamic environments. Robots that can adapt their bodies while deployed might be flexible and robust enough to meet this challenge. Previous work on dynamic robot morphology has focused on simulation, combining simple modules, or switching between locomotion modes. Here, we present an alternative approach: a self-reconfigurable morphology that allows a single four-legged robot to actively adapt the length of its legs to different environments. We report the design of our robot, as well as the results of a study that verifies the performance impact of self-reconfiguration. This study compares three different control and morphology pairs under different levels of servo supply voltage in the lab. We also performed preliminary tests in different uncontrolled outdoor environments to see if changes to the external environment supports our findings in the lab. Our results show better performance with an adaptable body, lending evidence to the value of self-reconfiguration for quadruped robots.

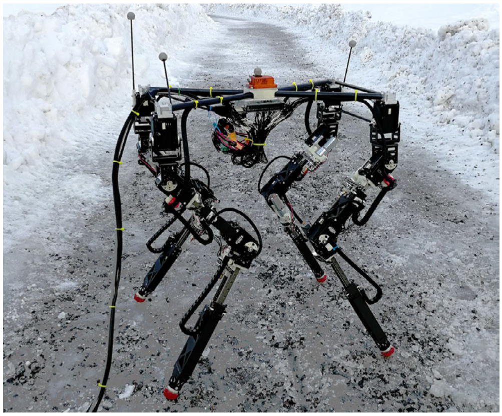


Figure 1: The robot during outdoor experiments. The legs can automatically change length, enabling experiments in self-reconfiguration.

1 INTRODUCTION

Robots are increasingly asked to operate in more dynamic and unpredictable environments, alongside other robots, or humans. The challenges of these environments can be handled through complex locomotion control or mechanical compliance [1], or by giving a robot the ability to adapt and learn. So far, robotic adaptation has focussed on a robot's control system, but adapting the body of a robot–its morphology–can provide a more fundamental flexibility [2]. The concept of embodied cognition suggests that the interaction of the mind, body, and environment can all contribute to the task solving ability of the robotic system [3]. Earlier work in evolutionary robotics has also shown that different morphologies emerge for environments of varying complexity [4]. Thus, a robot should adapt its body as well as its control system, to the environment and the task at hand.

In this paper, we introduce a practical four-legged robot including self-reconfigurable legs (Fig. 1). Each leg features three rotational joints that are used for locomotion, and two prismatic joints. While the prismatic joints are too slow to use in locomotion, they can actively alter the morphology of the robot during operation. This ability might be applied to adapt to a dynamic environment; alternatively, the body can be changed during traditional single-morphology experiments to validate solutions on different robot bodies.

With this robot, we wish to investigate whether adapting morphology can lead to an advantage in tackling dynamic situations or environments. This research goal can be summarised as the following hypothesis: *No single robot morphology performs best for all situations, tasks or environments*. That is, robots with dynamic morphology will be able to perform better than static morphologies, by modifying their own body in the face of changing situations.

In this paper, we present evidence supporting this hypothesis for our robot. In the lab, we examined how different morphologies with hand-tuned gaits perform when the torque of the servos is changed. This is done by constraining the robot's supply voltage, which emulates in-the-field depletion of a robot's battery. We also describe preliminary tests of the robot under battery power in two field environments to explore our results in more difficult environmental conditions. Our lab experiments support our hypothesis, and the field tests support our findings in the lab.

Contribution: There are two significant contributions in this paper. First, we demonstrate and introduce a practical robot system for researching self-modifying morphology. This system has been released as a fully certified open source hardware project, and can be used freely by other researchers. Secondly, we show through experimental results–both from the lab, and from field scenarios–that having a self-reconfigurable morphology helps our robot to maintain optimal performance when adapting to changing supply voltages and external environments. These experiments indicate that self-reconfigurable legs could improve the performance of robots doing complex tasks in dynamic environments.

2 Background

Being able to use different modes of locomotion will allow a robot to adapt to the most appropriate way of travel in dynamic and unknown environments. Some robots are able to change their locomotion mode without morphological change [5], while others change it by switching between separate structures, such as wheels and flight rotors [6]. Structures can also be shared by reusing parts of the body for different modes [7], which saves weight at the expense of mechanical and control complexity. In general, these robots have discrete morphologies used for each mode of locomotion, precluding adaptation from a continuous range of different morphologies in response to internal or external factors. Other robots can change parts of their bodies through adjustable compliance mechanisms [8], but these typically results in a much smaller impact on locomotion capability.

Modular self-reconfigurable robot systems, including complex simulations and physical implementations, can be divided into three architectures [9]. The simplest architecture is the chain or tree architecture, with a serial connection between modules. Zykov et al. [10] review many examples of systems that use this simple architecture, but still manage to show reasonably complex configurations. The lattice architecture has modules connected in parallel along a two or three-dimensional grid. This allows for more advanced base architectures, and connecting sub-parts of the system into meta-modules can yield interesting possibilities when changing the morphology of the system [11]. Other modular robots follow a mobile architecture, and can take on either of the previous architectures, or work as separate units. This is closely related to the field of swarm robotics; physically connecting a swarm of robots to form new, cooperative morphologies can yield very flexible solutions [12]. Despite these advances, modular robots still have a very coarse granularity when it comes to its morphology, when compared to other areas that change a robot's body.

The field of evolutionary robotics uses techniques from evolutionary computation to optimize control and–less often–morphology. Evolution of both control and morphology together is usually performed in simulation and presents additional challenges due to the complex search-space [13]. The difference between performance in a simulator and a real-world counterpart is referred to as the *reality gap*, and often makes it very challenging to transfer a result to the real world. A lot of interesting research has been done in simulation alone, but there are many reasons to move more of the research into hardware, as described in one of three grand challenges to the ER field posed by Eiben [14]. There are some examples of evolution of morphology in hardware, but these require either excessive human intervention [15], or use slow external reconfiguration of modular systems [16]. There are also examples of self-reconfiguring morphology used exclusively to guide the search for a better controller of a single hand-designed optimal morphology [17]. In our previous work, we have demonstrated in that earlier versions of the DyRET platform, presented in this paper, can be used for evolutionary experiments to optimize morphology using mechanical self-reconfiguration [18]. Our work was the first example of such an approach as far as we are aware.

3 System overview

Our robot was developed to be a platform for experiments on self-adaptive morphologies and embodied cognition, and is shown in Fig. 2. It is a certified open source hardware project,

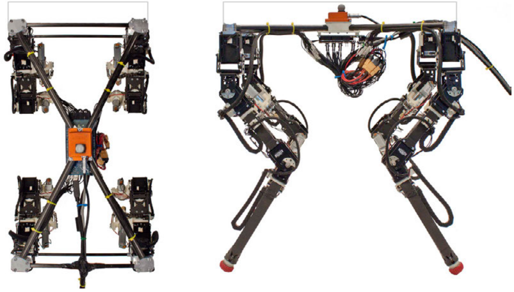


Figure 2: Top and side view of the robot, with measurements. The height of the robot is dependent on pose and leg length, and is typically between 600mm and 700mm in normal operation.

and documentation, code and design files are freely available online [19]. It can actively reconfigure its morphology by changing the lengths of the two lower links of its legs, the femur and tibia. The difference in height is illustrated in Fig. 3. Since it is used with machine learning techniques, the robot must withstand falls and unstable gaits, making maintainability and robustness important design factors.

Changing the length of the legs moves the center of gravity in the robot, affecting the balance. Longer legs also mean lower servo rotational velocity for a given end-effector path, at the expense of higher torque requirements. The length of the legs can therefore be used to mechanically gear the motors, and allow the robot to change where it sits in the trade-off between movement speed and force surplus continuously and autonomously.

3.1 Mechanics

The robot applies a mammal-inspired quadruped configuration. All parts can either be bought as relatively inexpensive commercial off-the-shelf components, or be printed on consumer-grade 3D printers. The parts for the robot without sensors are estimated to be about 6500USD in 2018, including 4300USD for the 12 servos alone. Some parts can optionally be made in aluminium for improved robustness, which is relevant if the robot is used for gait learning experiments. The main body of the robot is constructed with carbon fiber tubing of different diameters, which ensures a stable but low weight base for the four legs. The complete robot weighs 5.5kg, and operates tethered during all experiments.

The robot has four legs with five degrees of freedom each. The coxa (hip), femur (top leg), and tibia (lower leg) are all connected to revolute joints like traditional mammal robots, in addition to two prismatic joints to allow self-modification of the leg lengths. Each leg includes three Dynamixel MX-64AT servos, with integrated PID controllers that receive angle commands over USB. Off-the-shelf aluminium brackets are used to connect the servos to the rest of the robot where possible, with remaining connections using custom 3D printed and machined aluminium parts.

The two lower links of each leg, femur and tibia, can be reconfigured to different lengths. The reconfiguration mechanics is shown in Fig. 4. These linear actuators consist of small highly

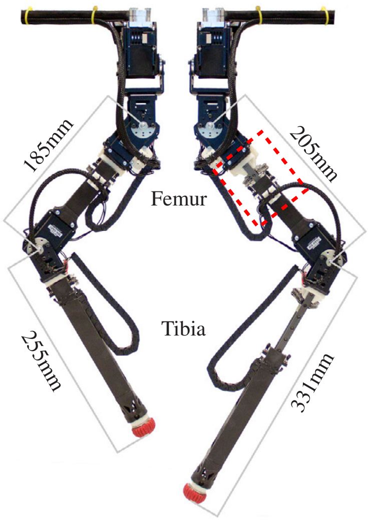


Figure 3: Diagram of the legs in their rest pose, showing the two lengths used in the paper. The shortest available length is to the left, and 80% of available length to the right, as used in our experiments. See Fig. 4 for details on the reconfiguration mechanism in the red square.

geared DC motors connected to lead screws through roller chain. The leg is connected to the lead screw with a self-lubricating plastic nut, and rides on aluminium rails by two carriages. The length of the leg is sensed by the encoder, which is calibrated on power-up using the mechanical end-stop. Leg cabling has been run through cable carriers that keep the cable runs constant regardless of leg length, along with cable lacing techniques to secure the cables with minimal strain. The low linear actuation speed (≈1mm/s) makes it ineffective to use this mechanism actively during the gait, so it is exclusively used for changing the morphology configuration.

3.2 Electronics

All twelve servos are connected to an external computer running the software through a USB serial adapter. Angular positions of all servos are reported to the system at approximately 60Hz, and new angle commands are received at the same rate. Temperature, current and load are also read, to ensure the servos stay within operating specifications. Servo position control is achieved using integrated PID-control in each servo.

The length of the reconfigurable legs are controlled using an Arduino Mega 2560 Rev 3 board with a custom PCB shield, which communicates with the software system through USB at 10hz. Limit switches are routed directly to the digital inputs of the microcontroller with internal pull-ups, and all encoders are connected directly to the analog inputs. The custom shield has twelve H-bridges to drive the DC motors in the linear actuators, which are controlled by PWM from the microcontroller. Since we are using a screw mechanism for the linear actuators with inherently high holding load and friction, a proportional controller for each prismatic joint is sufficient to achieve stable actuation with 0.5mm accuracy.

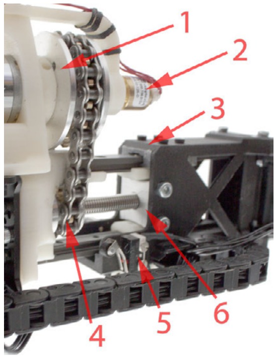


Figure 4: The reconfiguration mechanism, with the rest of the leg extending to the right of the image. (1) Brushed DC motor and sprocket, (2) Encoder for positioning, (3) Aluminium rail and carriages, (4) Threaded rod and sprocket, (5) Limit switch for zeroing, (6) Nut fixed to the movable part of the leg

An Xsens MTI-30 attitude and heading reference system (AHRS) is mounted close in the middle of the body to measure linear acceleration, rotational velocity and magnetic fields, giving data on absolute orientation at 100Hz. Reflective markers are mounted on the main body of the robot to allow motion capture equipment to record the position and orientation of the robot at 100Hz. The robot can carry enough weight to accommodate a full sensor package, such as a LIDAR and a depth camera, as well as an on-board Intel NUC computer.

3.3 Software

All software functions are implemented as separate Robot Operating System [20] nodes in C++, and an overview of the system with its main nodes is shown in Fig 5. An experiment manager node takes input from the user, and runs the different experiments. Trajectories with distance to move, direction, and configuration are sent to a trajectory controller that interfaces to the gait controller. Several different gait controllers can be used, as switching out nodes are simple plug-and-play procedures that can be done during system operation. The gait controller either sends commands to the hardware in the real world, or to the Gazebo simulator [21]. Feedback on performance is received by the gait evaluator from either simulations or the real world, and is analyzed, logged, and reported back to the experiment manager node.

3.4 Control

We have successfully implemented and used both high-level and low-level gait control. Only the high-level control is used for experiments with self-reconfiguration, as the gaits produced are more robust and easier to change for an engineer than low-level gaits. Low-level control is detailed in [22], and not used for the experiments in this paper.

The high-level control is an inverse-kinematics based position controller for the legs of the robot, making it easy for an engineer to hand design a gait, as well as to intuitively understand gaits that have been optimized by machine learning algorithms. It

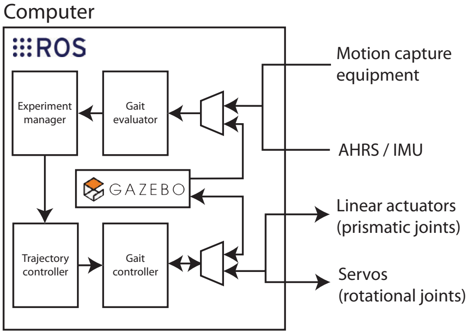


Figure 5: Overview of the software system. Each named black square is a node, and the system can either be connected to the Gazebo simulator, or to the physical robot and sensors in the real world.

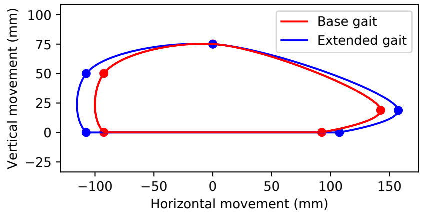


Figure 6: The leg path and control points for the gaits used in our experiments.

generates a continuous, regular crawl gait, and the body moves at a constant forward speed during the gait sequence, lifting each leg separately to maximize stability. The gait controller uses 8 parameters to generate points along an interpolating looping cubic hermite spline, seen in table 1. Ground height is given in millimetres (H_{ground}), and is dependent on the femur length (L_{femur}) and tibia length (L_{tibia}), following this equation:

$$H_{\text{ground}} = -430 - ((L_{\text{femur}} + L_{\text{tibia}}) * 0.8); \tag{1}$$

The 5 control points (x,y,z) for the spline are derived from step length (L_{step}), step height (H_{step}), step smoothing (S), and ground height (H_{ground}):

$$\begin{aligned}
&(0, \frac{L_{\text{step}}}{2}, H_{\text{ground}}) \\
&(0, \frac{-L_{\text{step}}}{2}, H_{\text{ground}}) \\
&(0, \frac{-L_{\text{step}}}{2}, H_{\text{ground}} + \frac{H_{\text{step}}}{1.5}) \\
&(0, 0, H_{\text{ground}} + H_{\text{step}}) \\
&(0, \frac{L_{\text{step}}}{2} + S, H_{\text{ground}} + \frac{H_{\text{step}}}{4})
\end{aligned} \tag{2}$$

A balancing wag movement (W) is added to allow the robot to lean to the opposite side of the leg it is currently lifting to allow for statically stable gaits. This is added to the position from the spline at each time step (t). Period (T) is calculated from the gait frequency parameter (f), while phase offset (W_ϕ) and amplitudes (W_{Ax} and W_{Ay}) comes directly from the gait parameters. An offset of 0.43 is added to offset forward and sideways movement.

$$\begin{aligned}
W_x &= \frac{A_x}{2} * tanh(3 * sin(\frac{2\pi * (t + (W_\phi * T))}{T})) \\
W_y &= \frac{A_y}{2} * tanh(3 * sin(\frac{2\pi * (t + (W_\phi + 0.43) * \frac{T}{2})}{\frac{T}{2}}))
\end{aligned} \tag{3}$$

We have also added a parameter for a lift duration (D_{lift}) to control what percentage of the gait period is used to lift the leg back to the front.

4 Experiments and results

We tested DyRET to find whether changing morphology really can deliver an advantage in different environments. We designed experiments in different environments that we speculated would require different leg lengths. In these limited number of environments, our hypothesis can be falsified if one morphology is the best performer in all situations. There are many measures of performance that could be used, but we chose to only look at forward speed, as it is both simple to measure and understand. By studying how different morphologies impact performance, we can gain an indication of whether mechanical self-reconfiguration could be useful in real-world dynamic environments.

Our experimental procedure was as follows. First, we hand designed the gaits to be used in the experiments by choosing conservative gait parameters. The main experiment is done in the lab to investigate how reducing the supply voltage, and thereby the torque of the motors, affects the performance of two different morphologies. The lab environment gives us stable evaluations, and the results should be directly transferable to real-world applications where the servos run on unregulated battery power. Then, to evaluate the plausibility of the lab experiments, we perform preliminary evaluation of the gaits in two different field environments: an indoor garage facility, and an outdoor footpath in winter conditions.

We used two morphology configurations for the experiments: one that uses the shortest available leg length (referred to as ”short robot”), and one that uses 80% of the available length (referred to as ”tall robot”). The maximum length of the legs was designed with optimal lab conditions in mind, so we use only 80% of the available length of both links as to not strain the robot in the more demanding environments.

We hand designed parameters for a conservative gait (referred to as ”base gait”) based on experience from previous experiments [18, 23]. The walking speed of the robot is limited by the maximum rotational speed of the servos. This speed is a function of torque, but we have selected a maximum allowable rotational speed of 25RPM in our current setup, based on specifications and experience. Since the legs of the tall robot are longer, the same leg endpoint movement requires a smaller rotational change. This means that the taller robot can walk faster than the shorter robot, given the same rotational speed limit. We therefore included a faster gait (referred to as ”the extended gait”) that could only be used on the taller robot without exceeding servo specifications. This gait has both increased frequency and step length, while all other gait parameters are kept the same, as seen in Table 1. The spline path for each gait can be seen in Fig. 6. The base gait is evaluated on both morphologies, while the extended gait is only valid for the tall robot.

Table 1: Hand designed gait parameters

Parameter	Symbol	Base gait	Extended gait
Step length	L_{step}	185mm	215mm
Step height	H_{step}	75mm	75mm
Smoothing	S	50mm	50mm
Frequency	f	0.275hz	0.35hz
Lift duration	D_{lift}	20%	20%
Wag phase	W_{ϕ}	0.0	0.0
Wag amplitude x	W_{Ax}	15mm	15mm
Wag amplitude y	W_{Ay}	10mm	10mm

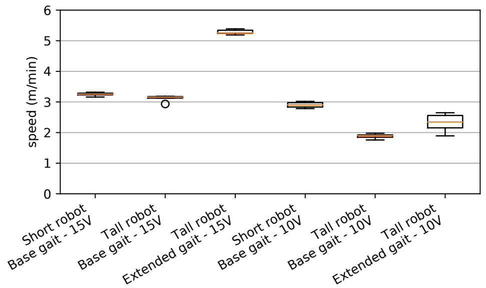


Figure 7: Results of the lab evaluations of the three different gait and morphology pairs at the different supply voltages.

4.1 Lab experiments

In the lab experiments, we change the supply voltage of the servos to investigate whether different leg lengths are needed when the torque available to the robot changes. Each gait and morphology pair was evaluated ten times with two different supply voltages by the robot walking 1.5m forwards and then 1.5m in reverse. This takes up to about 60s, depending on the gait speed. Pair-wise Mann-Whitney U tests with Holm correction were performed to assess statistical significance of differences.

The results are shown in Fig. 7, and with more details in table 2. Using the base gait at the higher voltage, we see that the short and tall morphology perform similarly at just over 3m/min. The tall robot with extended gait (Mdn = 5.26) perform significantly better than both the short robot (Mdn = 3.26, U = 0, $p < 0.001$), and the tall robot with the base gait (Mdn = 3.16, U = 0, $p < 0.001$).

At the lower voltage, we see that the short robot has a slight decrease in performance to just under 3m/min, while the tall robot with the base gait is now unable to match that speed, with a reduction to below 2m/min. The short robot (Mdn = 2.90) now outperforms the tall robot with both base gait (Mdn = 1.90, U = 0, $p < 0.001$) and extended gait (Mdn = 2.35, U = 0, $p < 0.001$).

For the lab experiments, we see that at the high voltage, the tall morphology performs best, with its extended gait. At the lower voltage, the short morphology performs best.

4.2 Field experiments

The field experiments in the garage and on the footpath, involved changes in the external environment, including surface friction, texture, temperature and humidity, to see if this affects the performance of different morphologies. In both field environments, the robot was powered by an external three-cell LiPo battery pack (11.1V) and controlled from a tethered laptop. The garage environment had a smooth concrete floor, with much lower friction than the lab's carpet, and ambient temperature of around +4°C. The outdoor evaluations were held on a footpath in Norwegian winter conditions (around −5°C) where the surface was a mix of compacted snow, ice, and gravel – a very challenging environment to retain traction, shown in Fig. 1. Three combinations of morphology and gait were evaluated by 120 seconds of forward walking in each environment. Each evaluation was replicated twice in the garage and three times on the outdoor footpath. Speed was evaluated by using a hand-held laser distance measurer along with a time measurement from the test program. The field experiments are only done as a preliminary investigation to see if our lab experiments are feasible also for external environment changes. We therefore have a limited number of evaluations, and are not able to do Mann-Whitney U tests to analyze statistical significance.

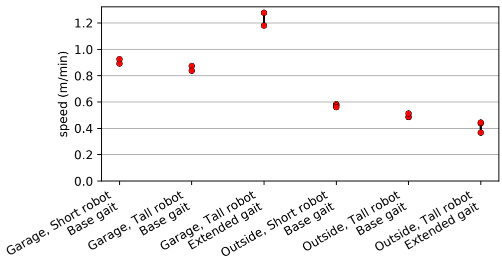


Figure 8: Results from the field experiments. Note that each garage evaluation has two data points, while the outdoor evaluations have three.

Table 2: Experiment results details

Experiment	Evals	Morphology	Gait	Range	Mean
Lab, 15V	10	Short	Base	[3.161, 3.319]	3.252
	10	Tall	Base	[2.936, 3.186]	3.135
	10	Tall	Extended	[5.188, 5.388]	5.283
Lab, 10V	10	Short	Base	[2.784, 3.019]	2.902
	10	Tall	Base	[1.759, 1.977]	1.884
	10	Tall	Extended	[1.893, 2.647]	2.326
Garage	2	Short	Base	[0.892, 0.926]	0.909
	2	Tall	Base	[0.838, 0.874]	0.856
	2	Tall	Extended	[1.181, 1.278]	1.229
Outside	3	Short	Base	[0.561, 0.583]	0.572
	3	Tall	Base	[0.486, 0,513]	0.495
	3	Tall	Extended	[0.368, 0.444]	0.416

The results from the field experiments are shown in Fig. 8, with details in table 2. Results from the garage show a big reduction in performance from the lab; the base gaits achieve a speed of about 0.9m/min, while the tall robot with the extended gait achieves a speed of about 1.2m/min. Although there is a reduction in speed when compared to lab conditions, we see the same trend as with the high voltage experiment in the lab: both morphologies perform similarly using the base gait, while the tall morphology with the extended gait walks faster.

On the footpath, all speeds are further reduced; and we see the same trend as observed with the lower voltage experiment in the lab. The shorter robot now outperforms the tall robot with both the extended gait, as well as base gait.

Our field experiments indicate that for the less demanding garage environment, the tall morphology performs best with its extended gait. In the more demanding outdoor environment, the short morphology performs best.

5 Discussion

The fact that the tall robot performed best at high supply voltage, and the short robot performed best at lower supply voltage, supports our hypothesis in lab conditions. This also strengthens our assumption that the self-adaptive legs can be used to adapt the robot to the new supply voltage by selecting different trade-offs between speed and torque.

In our preliminary indoor field experiments, we observed that the tall robot with the extended gait outperformed the short robot. In the more demanding outdoor environment, we saw that the short robot now outperformed the tall robot with either gait. This suggests that the trade-off between speed and torque in the shorter robot suits this new and demanding environment better than the taller morphology. The field experiments supports our findings in the lab, and suggests further exploration could be beneficial.

Our field experiments in Fig. 8 showed a reduced performance for all individuals when compared to the same gaits in lab conditions in Fig. 7. A different surface, temperature, control computer, and running on battery are all factors that could have contributed to this, and we observed slipping and stumbling of the robot that we had not previously seen in the lab.

The fact that we are working on a physical robot system has severely limited the number of samples we have used in our experiments. It is challenging to do longer experiments in outdoor environments, where there is a large number of variables that can not be controlled compared with lab experiments. The observed variance, however, was quite low, and statistical significance has been assessed where possible to help support our conclusions.

We are only able to address the hypothesis for our own robot, and the specific morphologies and environments in our experiments. We believe the main reason for the observed difference in performance between voltages or environments comes from the leg length gearing the motors and allowing different speed/torque tradeoffs. This effect would act linearly, and we do not expect to find new morphologies that outperform those we used in all our selected environments. Our results are encouraging, and suggest that other robots with self-reconfigurable hardware might derive similar advantages from adapting their bodies as we have.

6 Conclusion and future work

In this paper we introduced a novel four-legged mammal-inspired robot with mechanical self-modifying morphology. We hypothesise that no single robot morphology performs best for all situations, tasks or environments. To address this for our robot we ran lab experiments showing that different servo torques require different morphologies to perform well. We also performed preliminary field testing of the robot in two outdoor environments, which supported the results of our lab experiments. These results indicate that mechanically self-modifying robots may perform better in dynamic environments by adapting morphology as well as control to new conditions.

Even though we have shown clear indications that different morphologies are optimal in different situations or environments, we have yet to investigate how to switch between these or utilize a library of gaits and morphologies efficiently and autonomously. Doing more extensive field experiments with more evaluations and better tailored test setups would allow investigating how to automatically and continuously adapt control and morphology. We would also be able to investigate optimal morphologies for different environments, and the close ties and interactions between the environment and a robot's control and morphology. We evaluated the robot in two different outdoor environments, one of which was very challenging for the robot. It would be interesting to do more realistic experiments in a more extensive collection of environments, and introduce dynamic elements such as other robots or humans that might also affect the efficiency of different morphology-controller pairs.

We also hope that these experiments inspire more research on real world mechanical reconfiguration, and that our newly developed and open sourced platform might help lower the initial investment needed to begin such research by allowing others to use or extend our robot design [19], either in simulation or the real world.

7 Acknowledgements

This work is partially supported by The Research Council of Norway as a part of the Engineering Predictability with Embodied Cognition (EPEC) project, under grant agreement 240862, and through its Centres of Excellence scheme, project number 262762.

References

[1] P. Eckert, A. Spröwitz, H. Witte, and A. J. Ijspeert, "Comparing the effect of different spine and leg designs for a small bounding quadruped robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA15)*, May 2015, pp. 3128–3133.

[2] S. Murata and H. Kurokawa, "Self-reconfigurable robots," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 71–78, March 2007.

[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. E. Auerbach and J. C. Bongard, "Environmental influence on the evolution of morphological complexity in machines," *PLoS computational biology*, vol. 10, no. 1, 2014.

[5] K. Peterson and R. S. Fearing, "Experimental dynamics of wing assisted running for a bipedal ornithopter," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS11)*, 2011.

[6] A. Kossett, R. D'Sa, J. Purvey, and N. Papanikolopoulos, "Design of an improved land/air miniature robot," in *2010 IEEE International Conference on Robotics and Automation (ICRA10)*, May 2010.

[7] L. Daler, J. Lecoeur, P. B. Hahlen, and D. Floreano, "A flying robot with adaptive morphology for multi-modal locomotion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS13)*, Nov 2013, pp. 1361–1366.

[8] B. Vanderborght, A. Albu-Schaeffer, A. Bicchi, E. Burdet, D. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, M. Garabini, M. Grebenstein, G. Grioli, S. Haddadin, H. Hoppner, A. Jafari, M. Laffranchi, D. Lefeber, F. Petit, S. Stramigioli, N. Tsagarakis,

M. V. Damme, R. V. Ham, L. Visser, and S. Wolf, "Variable impedance actuators: A review," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1601 – 1614, 2013.

[9] M. Yim, W. min Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 43–52, March 2007.

[10] V. Zykov, W. Phelps, N. Lassabe, and H. Lipson, "Molecubes extended: Diversifying capabilities of open-source modular robotics," in *IROS-2008 Self-Reconfigurable Robotics Workshop*, 2008, pp. 22–26.

[11] D. J. Christensen and K. Stoy, "Selecting a meta-module to shape-change the atron self-reconfigurable robot," in *2006 IEEE International Conference on Robotics and Automation (ICRA06)*, May 2006.

[12] F. Mondada, A. Guignard, M. Bonani, D. Bar, M. Lauria, and D. Floreano, "Swarm-bot: from concept to implementation," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS03)*, Oct 2003.

[13] 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 *2017 Applications of Evolutionary Computation (EvoStar17)*, G. Squillero and K. Sim, Eds. Springer International Publishing, 2017, pp. 825–836.

[14] A. E. Eiben, "Grand challenges for evolutionary robotics," *Frontiers in Robotics and AI*, vol. 1, p. 4, 2014.

[15] M. Jelisavcic, M. de Carlo, E. Hupkes, P. Eustratiadis, J. Orlowski, E. Haasdijk, J. E. Auerbach, and A. E. Eiben, "Real-world evolution of robot morphologies: A proof of concept," *Artificial Life*, vol. 23, no. 2, pp. 206–235, 2017.

[16] V. Vujovic, A. Rosendo, L. Brodbeck, and F. Iida, "Evolutionary developmental robotics: Improving morphology and control of physical robots," *Artificial Life*, 2017.

[17] J. Bongard, "Morphological change in machines accelerates the evolution of robust behavior," *Proceedings of the National Academy of Sciences*, vol. 108, no. 4, pp. 1234–1239, 2011.

[18] 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 *2018 Genetic and Evolutionary Computation Conference (GECCO18)*. ACM, 2018, pp. 125–132.

[19] T. F. Nygaard, "DyRET GitHub repository." [Online]. Available: https://github.com/dyret-robot/dyret_documentation

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS04)*, vol. 3, Sept 2004, pp. 2149–2154 vol.3.

[22] 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. Cham: Springer International Publishing, 2018, pp. 719–733.

[23] 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)*, 2016, pp. 1–8.

Paper III

Real-World Evolution Adapts Robot Morphology and Control to Hardware Limitations

Tønnes F. Nygaard, Charles P. Martin, Eivind Samuelsen, Jim Torresen, Kyrre Glette

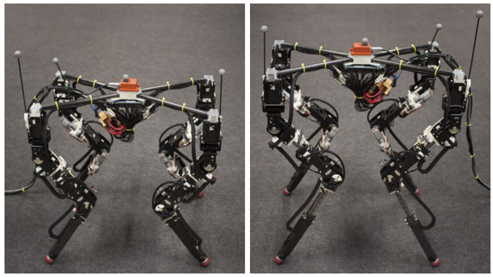
2018 Proceedings of the Genetic and Evolutionary Computation Conference
ISBN: 978-1-4503-5618-3, DOI: 10.1145/3205455.3205567

III

Real-World Evolution Adapts Robot Morphology and Control to Hardware Limitations

Tønnes F. Nygaard, Charles P. Martin, Eivind Samuelsen, Jim Torresen and Kyrre Glette
The Department of Informatics, University of Oslo

Abstract – For robots to handle the numerous factors that can affect them in the real world, they must adapt to changes and unexpected events. Evolutionary robotics tries to solve some of these issues by automatically optimizing a robot for a specific environment. Most of the research in this field, however, uses simplified representations of the robotic system in software simulations. The large gap between performance in simulation and the real world makes it challenging to transfer the resulting robots to the real world. In this paper, we apply real world multi-objective evolutionary optimization to optimize both control and morphology of a four-legged mammal-inspired robot. We change the supply voltage of the system, reducing the available torque and speed of all joints, and study how this affects both the fitness, as well as the morphology and control of the solutions. In addition to demonstrating that this real-world evolutionary scheme for morphology and control is indeed feasible with relatively few evaluations, we show that evolution under the different hardware limitations results in comparable performance for low and moderate speeds, and that the search achieves this by adapting both the control and the morphology of the robot.



(a) Shortest possible legs (b) Longest possible legs

Figure 1: The robot used in this research features self-modifying legs. The length of the two lower limbs of all four legs can be set individually with sub-millimetre accuracy.

1 Introduction

Evolutionary robotics (ER) uses techniques from evolutionary compution to optimize robot control and morphology, and aims to produce robots that are both robust and adaptive [1]. One of the biggest challenges in ER, is making the leap from software simulations to experiments evolving real physical robots [2]. Most ER research is done exclusively in simplified physics simulators [3]. Projects that transfer evolutionary results to physical robots often face discrepancies in performance between the simulator and the real world, referred to as the *reality gap*. Evolving in hardware on a real robot bypasses the problem of the reality gap completely, and can even be used for on-line adaptation of the system in its intended environment [4]. Many researchers do not use ER with the intent of producing an optimal robot controller or morphology, but to investigate evolutionary processes. Real world evolution might, for this objective, yield more realistic results since it exhibits the same noise and unpredictability that other physical systems in nature present. Evolving in hardware also lets us more closely investigate the embodied cognition aspect of robotics, namely how the interactions between mind, body, and environment affect how a robot solves a task. One of the biggest challenges in evolving in hardware today, is the long evaluation time required. This will be reduced with quicker and more accurate evaluation methods, and new production techniques allowing more systems to be run cheaply and efficiently in parallel might offset much of the difference between simulation and real world evaluation we see today.

In this paper, we investigate the extent to which control and morphology can be adapted by a real-world evolutionary system if the physical conditions of the system change. To achieve this, we use a four-legged robot with high-level control and self-reconfigurable morphology in the form of legs with motorised length adjustment, shown in Figure 1. In our investigations, we evolve the control and morphology of the robot at two different supply voltages, and compare the resulting individuals. Introducing a change in hardware conditions by turning down the supply voltage reduces both the available speed and torque of all joints by about 20%. A reduction in supply voltage would happen naturally to robots with motors directly powered by a depleting battery. Lower torque or speed can also be caused by internal effects like the temperature of the DC motors or wear and tear on the servo gears, or by external effects such as friction or texture of the walking surfaces, or the weight of the robot's payload. We also evaluate individuals resulting from the optimal voltage evolutionary run at the reduced voltage, to investigate the reduction in performance and need for adaptation to this limitation.

The results show that lowering the supply voltage of the robot–when it was evolved for the optimal voltage–can significantly impact the performance, with a reduction of 38% and 17% to speed and stability respectively. However, under evolutionary optimization at the reduced voltage, the robot is able to achieve comparable performance at low and moderate speeds to the optimal voltage individuals. We observe significant changes in both control and morphology between the two groups of individuals to achieve this.

The contribution of this paper is twofold: First, we demonstrate that evolution finds different morphology and control combinations suitable for our different hardware limitations, entirely by real-world evolution on a robot with self-reconfigurable morphology. Secondly, we demonstrate that by having a stable platform with high-level control, it is possible to do exploratory multi-objective morphology and control evolution in relatively few evaluations entirely in hardware. This allows us to investigate complex real-world dynamics not seen in ER experiments relying solely on software simulations.

2 Background

This section reviews related work in the area of evolutionary robotics, with a focus on real world evolution and evolution of morphology.

2.1 Evolutionary robotics

Modern specialized robots can be hard to develop, and are often designed by a team of engineers at considerable expense. Alternatively, in ER, robot controllers and morphologies can be designed or optimized automatically using evolutionary algorithms to identify new solutions [5]. In general, evolutionary design has been used to optimize a robot's control or morphology in an off-line fashion, before production, and in a different environment than where the robot would be working [4]. The method of embodied evolution uses on-line evolution of robots in the environment where they will be deployed, and thus the robots will be able to react to changes in that environment as they work [6]. Embodied evolution has, however, almost exclusively been applied to the control of a robot, as very few robots are able to modify their own morphology during an experiment without considerable human intervention.

2.2 Real world evolution

Most ER experiments are not performed on physical robots, but on virtual representations in a simplified physics simulator [3]. Here, the number and speed of evaluations is only limited by the access to computational power, and thus such experiments can be performed much faster than their real-world equivalent. Not only are real-world experiments more expensive in terms of building and maintaining a robot, but there are challenges due to noise in measurements caused by the body of the robot, its dynamic environment, and the interactions between them [7]. These advantages make it easy to see the appeal of only using simulations for evolutionary robotics.

One of the biggest challenges with using simulation in evolutionary robots, however, is the reality gap - the discrepancy between measurements of performance in simulation and the real world [8]. Modern physics simulators have different trade-offs between speed and accuracy, and game-based physics engines often sacrifice accuracy for additional stability [9]. Even simulators not focused on efficiency or stability can exhibit accuracy that is too low to allow direct transfers of results to real world counterparts. There are multiple approaches to deal with the reality gap, including adding noise in simulation [10], doing most of the evolutionary search in software before doing the last part in hardware [11], or by making a model of the disparity between simulation and reality, and use this to guide the search [12]. Some of these techniques reduce the reality gap significantly, but the difference still makes it challenging to transfer results to the real world - especially as robots are used in more complex environments. Other techniques guide the search towards individuals in simulation with behaviors that perform closer to their real world counterparts, and this might successfully circumvent some of reality gap problem to the degree where a subset of solutions might be transfered directly to the real world [3]. This does, however, limit the results to the small subset of solutions that has accurate performance in the simulator, and the search might therefore be drawn towards simple behaviors without dynamic effects, that are easier to simulate.

Evolving in hardware bypasses the problem of the reality gap completely, and if evolution is performed on the unrestricted system in the environment where it will be serving, also bypasses the problem sometimes seen in simplified or limited experiments in hardware as well [13]. Evolution in hardware is most often done off-line to perform a one-time adaptation to a new task or environment, but can also been done constantly in an on-line fashion to continuously adapt to both slow and abrupt changes to the robot itself or its environment [4]. There are several sources in the real world that contribute to uncertainty and noise in measurements of performance, but these are in many cases connected to the, often very complex, interactions between the control, body, and environment. Being able to study the synergy between these and see how a robot is able to exploit them separately and together to solve a given task is not possible in a simplified physics simulator.

2.3 Evolution of robot morphology

Evolutionary robotics can be used to evolve morphology and adapt a robot's body to the task it is solving, and the environment where it is doing it. It can even make the evolution of control quicker, and result in more robust gaits [14]. The field of artificial life evolves virtual creatures, closely related to evolution of robot bodies, but is mainly concerned with the study of the biological processes behind the evolution, and experiments are not done with the intention of producing hardware versions of the resulting bodies [15]. Most work in evolving virtual creatures is done in simulation alone, one of the earliest examples being Sims' work evolving bodies represented by three dimensional boxes [16]. This has also been done in later work [8], and expanded to more advanced creatures [17], though there have been several challenges related to the scalability of these techniques [18]. Evolution of morphology in robotics has also mostly been done in simulation, though the models used are more realistic than the virtual creature counterparts, and the intention is most often to end up with results that could be transferred to the real world. There are many examples of work evolving the morphology of different types of robots, for instance wheeled robots [19], legged robots [20], or even soft robots [21]. Morphology can also be evolved in modular robotics [22], though this most often refers to changing the way static modules are assembled.

There are some examples of evolution of robot morphology in simulation, where a select few morphologies are transferred for testing in the real world, including both legged [23] and more non-traditional designs [24], but these require excessive human intervention for each morphology tested in the real world. There are examples of morphological evolution in hardware directly as well, but many require excessive human intervention to build and assemble new morphologies [25], use slow external reconfiguration of modular systems [26], or no mechanical reconfiguration at all [27]. There have been examples of real-world robot evolution with self-modifying morphology, but only using the dynamic body to speed up or improve the evolution of controllers for a single optimal body [14]. The authors are not aware of any examples of real-world evolution of both control and morphology for complex legged robots.

Table 1: Characteristics of the Dynamixel MX-64AT servos when powered at different voltages.

Parameter name	**12V**	**14.8V**
No load speed	63rpm	78rpm
Stall torque	6.0Nm	7.3Nm
Stall current draw	4.1A	5.2A
Stall power draw	49.2W	78.0W

3 Robot and evolutionary setup

In this section we present the physical robot and its control system, the evolutionary setup, and the physical test setup we use in our experiments.

3.1 The robot

A custom robotic platform (shown in Figure 1) was used for all experiments in this paper, and is currently under development at the University of Oslo. Details on the platform itself can be found in our previous work [28], and we have previously used it for evolving control with static morphology [29]. The top frame measures about 480mm by 300mm, connecting the four legs in a mammalian configuration. All legs have the ability to change their length, with a minimum length of 550mm, and maximum length of 670mm. The middle link, or femur, has a minimum length of 185mm and a maximum of 210mm, while the lowest link, tibia, has a minmum length of 255mm and a maximum of 350mm.

Each leg includes three Dynamixel MX-64AT servos, with onboard PID controllers to receive the angle commands over USB. These servos are powered at different voltages in the experiments, and their operating characteristics are shown in Table 1. Reducing the voltage from the optimal voltage at 14.8V to a reduced voltage of 12V limits both torque and control by about 20%.

The reconfigurable legs use small DC motors connected to lead screws, with aluminium rails for mechanical strength. An Arduino Mega with a custom shield is used for the control, and we achieve a sub-millimetre accuracy on the leg length. The low speed of reconfiguration ($\approx$1mm/s) makes it ineffective to use these actively during the gait, so they are exclusively used for changing morphology, and are not seen by the controller.

An Xsens MTI-30 *Attitude and Heading Reference System* (AHRS) is mounted close to the middle of the body to measure linear acceleration, rotational velocity and magnetic fields, giving data on absolute orientation at 100Hz. Reflective markers are mounted on the main body of the robot to allow motion capture equipment to record the position and orientation of the robot at 100Hz. The complete robot weighs 5.5kg, and operates tethered during all experiments.

3.2 Control system

We use a high-level inverse-kinematics based position controller for the legs of the robot. The platform also supports a low-level controller, but this is only used in simulation experiments, due to the high number of evaluations needed before stable gaits are found. A continuous, regular crawl gait [30] was chosen, where the body moves at a constant forward speed during the gait sequence, and lifts each leg separately to maximize stability. This setup allows gaits that are statically stable, although the low weight of the legs in relation to the body makes achieving faster gaits without introducing dynamic effects challenging. The path for each individual leg end is defined by a Catmull-Rom spline.

Table 2: Gait parameters. These have been constrained (*) to limit the robot to a maximum speed of 10m/min.

Category	**Name**	**Values**
Spline shape	*step_length*	[5mm, 300mm]*
	step_height	[25mm, 75mm]
	step_smoothing	[0, 50mm]
Gait timing	*gait_frequency*	[0.2Hz, 2Hz]*
	lift_duration	[5%, 20%]
Balancing	*wag_phase*	[-0.2, 0.2]
	wag_x_amp	[0, 50mm]
	wag_y_amp	[0, 50mm]
Morphology	*femur_length*	[0, 25mm]
	tibia_length	[0, 95mm]

The gait generator uses parameter ranges defined in Table 2 and generates a number of control points for the spline, resulting in a continuous gait path for each leg[1]. Three parameters are used for manipulating the control points. The parameter step_length controls the length of the ground contact line, while step_height determines the height of the step. The step_smoothing parameter regulates the angle of movement at the point where the leg hits the ground, by stretching out the front of the spline. This was added to allow for a reduction of the impact forces from each step, by making contact with the ground in a more horizontal direction.

To increase the stability of the gait, a configurable balancing "wag" movement was added where the robot leans to the opposite side of the currently lifted leg. This ensures a higher margin of stability, and is required for a statically stable gait due to the relatively high mass of the legs compared to the body. Parameters for the phase and amplitude of the balancing wag can be changed individually for the lengthwise and sideways movement.

The maximum theoretical speed of the robot is given by the gait_frequency and step_length parameters; however, the actual speed of the robot also depends on its stability, and friction between its feet and the ground. Setting a high gait_frequency and low step_length, and also a low gait_frequency and high step_length would result in valid gaits. If both parameters are set too high, however, the robot might end up damaging itself by trying to achieve a non-realistic forward speed. We therefore limit the product of step_length and gait_frequency to 10m/min. The lift_duration parameter decides how much of the gait period is used to lift the leg through the air, before beginning the next step.

The gait is made completely independent of the robot morphology by sending the goal position of the legs to an inverse kinematics function that reads the lengths of the legs at 10Hz. No adaptation of any kind is done in the controller for the different morphologies, as we do not want to impose any limitations based on a priori knowledge of the design. It might, for instance, be intuitive that an individual with longer legs might work better taking longer steps, but we do not want to add more dependencies between morphology and control than exists naturally in the system. Minimizing the dependencies makes it easier to analyse the results, as there are fewer factors affecting the evolutionary search and its results.

The control system is implemented in C++ and uses the software framework *Robot Operating System* (ROS) [31]. The leg end positions from the gait controller are sent through an

[1]Details on control point generation can be found in the source code at `http://robotikk.net/project/dyret/`

Table 3: Parameters for the evolutionary experiments

Name	Value
Algorithm	NSGA-II
Evaluation time	Maximum 60s
Parameters	Real: [0, 1]
Recombination	None
Mutation	Type: Gaussian
	Probability: 1.0
	Initial sigma: 1/6
	Sigma decay per generation: 0.05
	Minimum sigma: 0.05
Evaluations	Population: 8
	Generations: 8
	Runs per experiment: 3
	Evaluations per re-evaluation: 10

inverse kinematics function to obtain the angles necessary to achieve the specified pose. The different functions of the robot controller are implemented as individual ROS nodes, and run on a computer connected to the robot by cable.

3.3 Evolutionary setup

Mammal-inspired four-legged robots, as used in this work, are more prone to fall than spider- or lizard-inspired robots commonly used in evolutionary robotics. Our robot's narrow stance, downward extending legs, and high centre of gravity, present much more danger of falling to the side than other bio-inspired designs. To be able to evolve fast gaits that are also robust on our platform, it is important to include stability as a fitness objective, in addition to speed. These two goals are conflicting, as a robot standing still has great stability, while a fast robot necessarily has some movement that will be interpreted as instability. We therefore chose the NSGA-II algorithm [32] to identify a Pareto front of solutions; a number of gaits with different trade-offs between the two objectives. The software running the evolutionary algorithm uses Sferes2 [33], a C++ framework for evolutionary experiments.

Parameters are represented as real numbers with the values shown in Table 2. Gaussian mutation is used on all genes with an initial sigma of 1/6, which decays per generation to enhance the exploration early in the search, but still allow exploitation in later generations. These meta-parameters were tuned to perform well at the low number of evaluations used in our experiments. Since both exploration and exploitation is covered by the mutation, we use no recombination. The step_length and gait_frequency are further limited by a maximum theoretical speed of 10m/min. If after mutation the gait surpasses this limit, mutation is done again until it is within the limits. Three runs are done for each experiment, and they all contain 8 generations of 8 individuals each, for a total of 192 evaluations for each experiment. When re-evaluating single individuals, they are evaluated 10 times each to get a satisfactory statistical distribution of their fitness in the real world. To avoid effects on the performance due to setup, we did our re-evaluations on a different day than the original evolutionary runs. The evolutionary parameters are summed up in Table 3.

Two fitness functions are used in the experiments in this paper, speed and stability. The speed is calculated by using the duration of the gait and the Euclidean distance between the start and end position captured by the motion capture equipment, as seen in Equation 1, resulting in a measure of traversed meters per minute. We use a fitness function for stability based on the orientation and measured linear acceleration from the AHRS. The full stability objective function, seen in Equation 2, is a weighted sum of the linear acceleration and orientation function, where *acc* are samples from the accelerometer, *ang* are samples from the orientation output of the AHRS, *i* is the sample index, and *j* is the axis of the sample. The accelerometer records data in the x, y and z-axes, while orientation is recorded in roll, pitch and yaw. The scaling factor α was chosen to provide a balance between the two stability measurements by having acceleration and orientation affect the fitness value equally, and was in these experiments set to 0.02. The stability objective function is negated to allow for maximization of both objective functions, which means that a perfectly stable robot has a stability score of 0. Samples in both functions are recorded at 100Hz.

$$F_{speed} = \frac{||P_{end} - P_{start}||}{time_{end} - time_{start}} \tag{1}$$

$$G(A_j) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_{j,i} - \overline{A_j})^2}$$
$$F_{stability} = -\left(\alpha * \sum^{axes} G(Acc_{axis}) + \sum^{axes} G(Ang_{axis})\right) \tag{2}$$

3.4 Physical test setup and evaluations

The goal of the physical test setup is to maximize the quality of measurements, while minimizing down time and requirements for human intervention. Motion capture equipment is used to provide a precise and accurate reading of position for estimation of speed. The duration of each gait test is chosen to provide a good balance between the number and accuracy of evaluations, given the time budget. Each evaluation is obtained by walking one and a half meters forward, and then walking back to the start position using the same gait in reverse, before averaging the fitness values achieved in both directions. Each path is restricted by a timeout of 15 seconds, to limit the time spent on evaluating the slower individuals. Evaluating a gait both directions help cancel out any asymmetric dynamics in the system that is caused by minor differences in the mechanics of the left and right side of the robot.

Both the robot and control system are designed to ensure repeatability for gaits by keeping the distance moved between each evaluation minimal. This is achieved by having the robot sequentially lift and reposition the legs to the start pose of new gaits after each evaluation. Two walking sequences of 15 seconds, in addition to mechanical reconfiguration and repositioning of legs before and after the gait, results in a maximum of about 60 seconds used for each evaluation. Some human intervention is required if the robot falls, or gets too close to the perimeter of the experiment area. In practice, such intervention seems to be required every one to five minutes, depending on the objectives used and stage of evolution. If the robot falls or finishes evaluation without being parallel to the floor, the program pauses and waits for human intervention before continuing, to ensure only valid fitness scores are recorded.

4 Experiments and results

Our main experiment is comprised of evolutionary multi-objective runs at the two different voltage levels. We compare

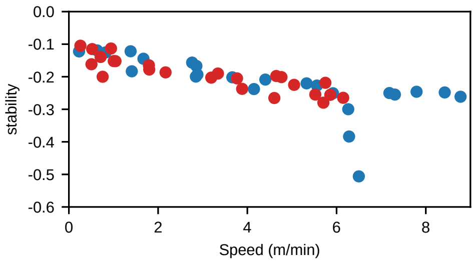
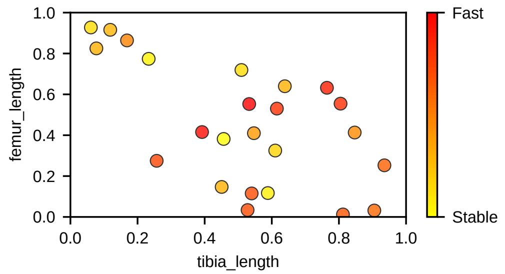
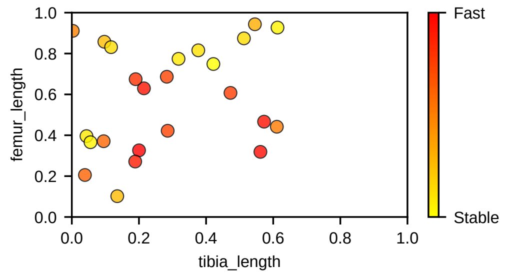


Figure 2: Comparison between fitness of the last generations evolved with optimal and reduced voltage. All individuals are optimized towards the top right, where an individual would have both high speed and stability. Blue is evolved at the optimal voltage, while red is evolved at the reduced voltage.



(a) Optimal voltage leg lengths.



(b) Reduced voltage leg lengths.

Figure 3: The length of the two reconfigurable leg segments for the last generations of the evolutionary runs. The colour indicates where in Fig. 2 the individual comes from, with the fastest robots in red, and the most stable robots in yellow.

the fitness from the two groups of runs, and examine the resulting individuals to identify signs of adaptation of control and morphology in the populations. A selection of individuals from the optimal voltage runs is then re-evaluated 10 times each to gain a representative measurement of their fitness. This re-evaluation is done at both optimal and reduced voltage to determine how the change in supply voltage affects performance, and to shed light on the need for adaptation when subjected to this change. In this section, we first present the results of the main experiment, before showing the results from the re-evaluation of individuals.

4.1 Evolutionary runs

The last populations of all runs are shown in Figure 2. The optimal voltage runs achieve a higher speed of up to 9m/min, while the reduced voltage runs achieve speeds of just over 6m/min. Even though only optimal voltage individuals achieve high speeds, the performance of both runs is comparable for small and moderate speeds. The final populations for both groups have a reasonably linear shape broken only by a low-stability tail at around 6m/min in one of the optimal voltage runs.

Figure 3 shows the morphologies that resulted from the runs with the two different voltages. The colour of the individuals shows the difference in fitness of the individuals, showing the relationship between morphology and achieved speed and stability in the experiments. For the optimal voltage individuals in Figure 3a, we see a smaller clustering of high femur length and low tibia length individuals, and a larger clustering of high tibia length with moderate to low femur length individuals. They use a maximum of 79% of the available reconfigurable leg length, while the mean individual uses 50% of its available reconfigurable length.

For the reduced voltage runs in Figure 3b, we see that individuals use the whole range of reconfigurable femur length, but only up to about 60% of the reconfigurable tibia length. Since the reconfigurable length of the tibia is much longer than for the femur, we only see up to 68% of the total available reconfigurable leg length being used, with a mean of about 35% for the reduced voltage. We also see from the graphs in Figure 3 that the performance of the individuals is not proportional to the total length of the robot, as several of the tallest robots only have moderate speeds, and a couple of the shorter individuals have some of the faster speeds.

The boxplot in Figure 5 reveals some of the differences in control and morphology parameters between the populations. There are clear differences in the tibia_length and wag_x_amp parameters, and moderate differences in femur_length, step_smoothing, and step_height. A detailed study of how each of the ten parameters is affected by the hardware limitations is out of the scope of this paper, so these differences are not investigated further individually.

However, we wish to analyse them on a group basis, in order to study the differences in morphology and control between the optimal and reduced voltage runs. To achieve this, linear discriminant analysis (LDA) was applied separately to the morphology and control parameters to give a one-dimensional representation of each group. This was followed by a Mann-Whitney U test to establish significance. The Mann-Whitney U test indicated a significant difference in the one-dimensional reduction of the two parameters for morphology, femur_length and tibia_length, due to the change in voltage ($U = 138$, $p < 0.01$), with Cliff's delta effect size of -0.52. The same analysis on the eight control parameters reduced to one, also indicated significant differences ($U = 92$, $p < 0.01$), with a Cliff's delta effect size of -0.68.

4.2 Re-evaluation of individuals

Since we are using a high-level controller, it can be hard to directly predict how a change to a robots internal or external

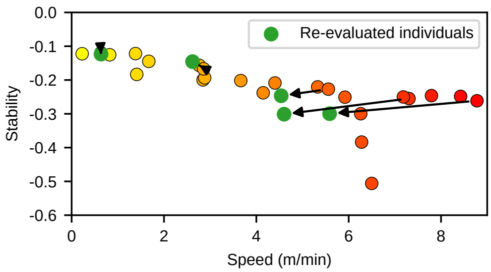


Figure 4: Fitness of individuals compared to their re-evaluations at reduced voltage. Green circles show the fitness at reduced voltage, and the black arrows show the change in fitness for these individuals.

Table 4: Means and standard deviations of results from the re-evaluation of selected individuals. The original fitness from the evolutionary run is included, in addition to re-evaluated fitness at both optimal and reduced voltage. (* Statistically significant difference)

Speed				**Stability**			
evo	*optimal*	*reduced*	*change*	*evo*	*optimal*	*reduced*	*change*
0.63	0.65 ± 0.02	0.64 ± 0.02	-1.9%	-0.11	-0.13 ± 0.00	-0.12 ± 0.00	+8.2%*
2.76	2.80 ± 0.03	2.62 ± 0.02	-6.5%*	-0.16	-0.16 ± 0.00	-0.15 ± 0.00	+6.4%*
5.56	5.54 ± 0.06	4.54 ± 0.24	-18.0%*	-0.23	-0.22 ± 0.01	-0.25 ± 0.01	-12.8%*
7.31	7.15 ± 0.13	4.60 ± 0.26	-35.7%*	-0.26	-0.26 ± 0.01	-0.30 ± 0.01	-14.9%*
8.78	8.96 ± 0.13	5.59 ± 0.17	-37.6%*	-0.26	-0.26 ± 0.02	-0.30 ± 0.01	-16.7%*

environment affects it, and we need to verify if adaptation is actually necessary when changing the voltage, or if the controller is able to handle both scenarios. For this, we chose five individuals with different fitnesses from the optimal voltage runs. These were then evaluated ten times at their original voltage, before being tested again at the reduced voltage. Re-testing under the original conditions is important to give an accurate comparison, as the noise in hardware measurements means that the single evaluation during evolution might not be representative of its true performance.

The results are summarized in Table 4. We can observe that for the two slowest individuals, the stability actually increases, while the stability decreases by 13% to 17% for the others. All mean speeds decrease, with the biggest reduction at 38%. All changes, except in the speed of the slowest individual, were shown to be statistically significant ($p < 0.01$) using the Mann-Whitney U test with Holm-correction of the p-values. Figure 4 shows the change in fitness from original to reduced voltage, where green circles denote re-evaluated individuals at the lower voltage. This figure reveals the large drop in speed for fast individuals particularly clearly.

5 Discussion

The decrease in performance seen in Figure 4 shows that lowering the supply voltage of the system affects the robot's gait. Reducing both torque and speed of the robot joints yielded a speed reduction of up to 38% and a stability decrease of up to 17%. This large discrepancy shows the need for adaptation to keep performing well in dynamic environments with changing hardware conditions. There is a large number of factors that can affect the performance of a robot, and it is likely that many robots, especially if working in complex environments or alongside other agents, might see similar, or even larger, differences in performance than we saw here. A robot can adapt to some of these factors using the evolutionary techniques shown in this paper, but they have not been tuned to respond quickly to abrupt changes, and are only meant as an off-line adaptation to new hardware limitations or environments.

We see from the difference in Figures 3a and 3b that the lower powered individuals are not able to exploit the full available length of the legs. This is supported by the fact that the mean reconfigurable leg length is 50% for optimal voltage runs, and only 35% for reduced voltage runs. Lower leg lengths can be seen as a gearing of the motors, as shorter legs trade speed for torque, and a reduction in leg lengths can therefore be seen as a response to the reduced torque. An interesting detail shown in Figure 5 is that even though results from the optimal voltage runs have a higher mean leg length, the femur_length is generally highest in the reduced voltage runs. Even though the interaction of these parameters under evolutionary optimisation is very complex, and might require more experiments to be understood fully, we still see a significant change in both morphology and control, which shows that the evolutionary search is able to adapt to the new hardware conditions by utilizing both.

The number of evaluations performed in this real-world study is limited compared to simulated ER research. Early experiments showed little to no improvement in fitness past the sixth generation, so we chose to do eight generations for a high probability of the search to converge. We also saw that the resulting populations contained a good number of individuals with different trade-offs between the different objectives, indicating that we had sufficient population size. Considering Figure 2, we see that there isn't a big difference in performance of the final populations between the runs, and we consider it unlikely that more runs would change the results considerably. Figure 5 shows a large diversity in final populations for the two groups of runs. We would expect to see much smaller variations for a converged evolutionary search with one objective, but that is not the case when doing multi-objective evolution using NSGA-II. This algorithm has a mechanism for maximising the fitness diversity in each front of the population, and since our two fitness objectives are conflicting, we end up with a range of different individuals with different trade-offs between these two objectives, which necessarily results in higher diversity in the populations as well.

Evolving robots in real-world environments is often challenging due to noise in measurements. The standard deviations in Table 4 showed only small variations of performance in our experiments, even when the re-evaluations was done on another day. These results confirm that we limited noise and uncertainty in our measurements to an acceptable level.

Figure 4 shows that only the faster individuals suffer significant losses in fitness and that more stable individuals are robust to the reduced supply voltage. Visual observation of the evaluations suggested that the reduction in performance is most likely caused by the lower stability. The theoretical speed of the gait is given by the high-level controller and the gait_frequency and step_length parameters, but unstable gaits stumble or miss steps, leading to lower distances covered in the same time. This indicates that if we are to deploy this robot in new conditions, it might be wise to select more stable gaits, as they are most likely more robust to unknown environments.

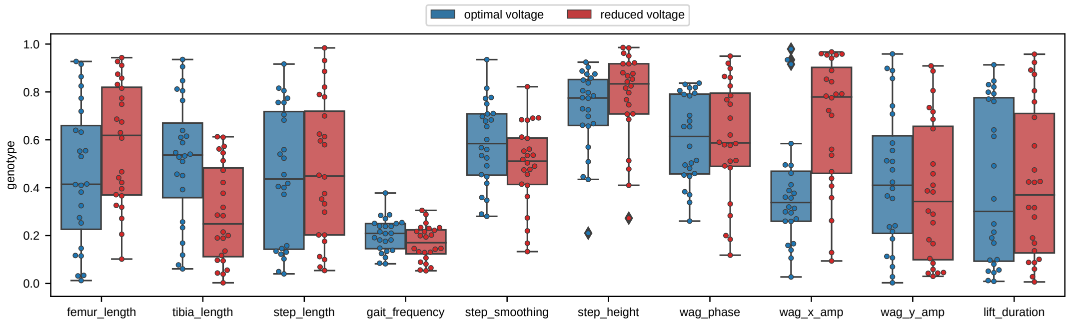


Figure 5: Genotype values and distributions for all individuals in the final generations resulting from the evolutionary runs.

6 Conclusions and future work

In this paper we investigated the effects of lowering servo torque and speed on evolved robots, and to what degree the robot through evolutionary techniques was able to adapt to this change. This large reduction in performance from lowering the voltage shows the need for adaptation to keep performing well in dynamic environments with changing hardware conditions. We showed that the evolutionary search was able to achieve comparable results to the original run at low and moderate speeds by changing both the control and morphology of the robot. We also demonstrated the feasibility of doing multi-objective exploratory morphology and control evolution entirely in hardware on our new platform.

An avenue for future expansion of this work would be to further investigate the actual contribution from using evolutionary algorithms over random search, and investigate other techniques from machine learning to implement on-line optimization as well. The adaptation to lower servo torque and speed in this paper has been done off-line, and we expect that doing this adaptation on-line instead would pose additional challenges with interesting solutions and results. Adding closed-loop control, opening up more parameters in the control system, or having separate parameters for each leg would give the system more possibilities for adapting, though getting feasible gaits in the start of the search with a mammal-inspired configuration can be very challenging. Current methods for generating behavioural repertoires could benefit from dynamic morphologies. It may also be possible to reduce the need for human intervention, allowing experiments in even more complex environments, encouraging investigations into embodied cognition and the interactions between robot body, mind, and environment.

We showed that our evolutionary system is able to adapt both control and morphology to new hardware limitations, but also that it is possible to do multi-objective exploratory morphology and control evolution in relatively few evaluations entirely in hardware, hopefully inspiring more researchers to take the leap into real world evolutionary experiments.

Acknowledgements

This work is partially supported by The Research Council of Norway as a part of the Engineering Predictability with Embodied Cognition (EPEC) project, under grant agreement 240862.

References

[1] 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.

[2] A. E. Eiben, "Grand challenges for evolutionary robotics," *Frontiers in Robotics and AI*, vol. 1, p. 4, 2014.

[3] 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.

[4] S. Doncieux, J.-B. Mouret, N. Bredeche, and V. Padois, "Evolutionary robotics: Exploring new horizons," in *New Horizons in Evolutionary Robotics*, S. Doncieux, N. Bredèche, and J.-B. Mouret, Eds. Springer Berlin Heidelberg, 2011, pp. 3–25.

[5] S. Nolfi and D. Floreano, "Synthesis of autonomous robots through evolution," *Trends in cognitive sciences*, vol. 6, no. 1, pp. 31–37, 2002.

[6] R. A. Watson, S. G. Ficici, and J. B. Pollack, "Embodied evolution: Distributing an evolutionary algorithm in a population of robots," *Robotics and Autonomous Systems*, vol. 39, 2002.

[7] F. Silva, M. Duarte, L. Correia, S. M. Oliveira, and A. L. Christensen, "Open issues in evolutionary robotics," *Evolutionary computation*, vol. 24, no. 2, pp. 205–236, 2016.

[8] N. Lassabe, H. Luga, and Y. Duthen, "A new step for artificial creatures," in *2007 IEEE Symposium on Artificial Life*, April 2007, pp. 243–250.

[9] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4397–4404.

[10] 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.

[11] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada, "How to evolve autonomous robots: Different approaches in evolutionary robotics," *Artificial life IV: Proceedings of the 4th International Workshop on Artificial Life*, pp. 190–197, 1994, r. A. Brooks and P. Maes (eds.).

[12] S. Koos, J.-B. Mouret, and S. Doncieux, *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 122–145, 2013.

[13] M. A. Gongora, B. N. Passow, and A. A. Hopgood, "Robustness analysis of evolutionary controller tuning using real systems," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 606–613.

[14] J. Bongard, "Morphological change in machines accelerates the evolution of robust behavior," *Proceedings of the National Academy of Sciences*, vol. 108, no. 4, pp. 1234–1239, 2011.

[15] J. Lehman and K. O. Stanley, “Evolving a diversity of virtual creatures through novelty search and local competition,” in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’11, 2011, pp. 211–218.

[16] K. Sims, “Evolving 3d morphology and behavior by competition,” *Artificial life*, vol. 1, no. 4, pp. 353–372, 1994.

[17] M. Komosinski, “The world of framsticks: Simulation, evolution, interaction,” in *Virtual worlds*. Springer, 2000.

[18] N. Cheney, J. Bongard, V. SunSpiral, and H. Lipson, “On the difficulty of co-optimizing morphology and control in evolved virtual creatures,” in *Proceedings of the Artificial Life Conference*, vol. 2016, 2016, pp. 226–234.

[19] W.-P. Lee, J. Hallam, and H. H. Lund, “A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, May 1996, pp. 384–389.

[20] 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 International Publishing, 2017, pp. 825–836.

[21] J. Rieffel, D. Knox, S. Smith, and B. Trimmer, “Growing and evolving soft robots,” *Artificial life*, vol. 20, no. 1, pp. 143–162, 2014.

[22] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, “Evolved and designed self-reproducing modular robotics,” *IEEE Transactions on robotics*, vol. 23, no. 2, pp. 308–319, 2007.

[23] E. Samuelsen, K. Glette, and J. Torresen, “A hox gene inspired generative approach to evolving robot morphology,” in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 2013, pp. 751–758.

[24] G. S. Hornby, H. Lipson, and J. B. Pollack, “Evolution of generative design systems for modular physical robots,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 4, 2001, pp. 4146–4151 vol.4.

[25] M. Jelisavcic, M. de Carlo, E. Hupkes, P. Eustratiadis, J. Orlowski, E. Haasdijk, J. E. Auerbach, and A. E. Eiben, “Real-world evolution of robot morphologies: A proof of concept,” *Artificial Life*, vol. 23, no. 2, pp. 206–235, 2017.

[26] V. Vujovic, A. Rosendo, L. Brodbeck, and F. Iida, “Evolutionary developmental robotics: Improving morphology and control of physical robots,” *Artificial Life*, 2017.

[27] L. Lichtensteiger and P. Eggenberger, “Evolving the morphology of a compound eye on a robot,” in *Advanced Mobile Robots, 1999. (Eurobot ’99) 1999 Third European Workshop on*, 1999, pp. 127–134.

[28] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, “Self-modifying morphology experiments with DyRET: Dynamic robot for embodied testing,” 2018. [Online]. Available: https://arxiv.org/abs/1803.05629

[29] 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)*, 2016, pp. 1–8.

[30] P. G. de Santos, E. Garcia, and J. Estremera, *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer Science & Business Media, 2007.

[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.

[33] J.-B. Mouret and S. Doncieux, “SFERESv2: Evolvin’ in the multi-core world,” in *Proc. of Congress on Evolutionary Computation (CEC)*, 2010, pp. 4079–4086.

Paper IV

Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds

Tønnes F. Nygaard, Charles P. Martin, Jim Torresen, Kyrre Glette

Lecture Notes in Computer Science book series, volume 11454
ISBN: 978-3-030-16692-2 , DOI: 10.1007/978-3-030-16692-2_41

This paper was originally published in a single-column format, but has been revised to fit in the two-column thesis template.

IV

Evolving Robots on Easy Mode: Towards a Variable Complexity Controller for Quadrupeds

Tønnes F. Nygaard, Charles P. Martin, Eivind Samuelsen, Jim Torresen and Kyrre Glette
The Department of Informatics,
University of Oslo

Abstract – The complexity of a legged robot's environment or task can inform how specialised its gait must be to ensure success. Evolving specialised robotic gaits demands many evaluations—acceptable for computer simulations, but not for physical robots. For some tasks, a more general gait, with lower optimization costs, could be satisfactory. In this paper, we introduce a new type of gait controller where complexity can be set by a single parameter, using a dynamic genotype-phenotype mapping. Low controller complexity leads to conservative gaits, while higher complexity allows more sophistication and high performance for demanding tasks, at the cost of optimization effort. We investigate the new controller on a virtual robot in simulations and do preliminary testing on a real-world robot. We show that having variable complexity allows us to adapt to different optimization budgets. With a high evaluation budget in simulation, a complex controller performs best. Moreover, real-world evolution with a limited evaluation budget indicates that a lower gait complexity is preferable for a relatively simple environment.

1 Introduction

Robots are used in more and more demanding and changing environments. Being able to adapt to new situations, unexpected events, or even damage to the robot itself can be crucial in many applications. Robots that are able to learn and adapt their walking will be able to operate in a much wider range of environments.

Selecting a suitable gait controller for a robot learning to walk can be very challenging, especially when targeting hardware platforms. A controller is often chosen early in the design process of a robot, and is used in a wide range of different evaluation budgets and environments. Simple controllers produce gaits with a limited diversity. More complex gait controllers are able to produce a wider range of gaits, with higher variance in performance and behaviors.

Controllers that are too complex might exhibit bootstrap problems, where the initial random population does not contain a suitable gradient towards better solutions [1]. Random solutions might also exhibit a high probability of the robot falling, making it more challenging to evolve in hardware. Another important factor is the larger and more complex search space, which might require more evaluations to converge than practically possible without simulations [2].

A controller can be made simpler by embedding more prior knowledge, for instance by reducing the allowable parameter ranges of the controller. When the size of the search space is reduced, fewer evaluations are needed, and with more conservative parameter ranges, falling can be greatly reduced. Reducing the gait complexity too much, however, leaves the system with a very narrow and specialized controller that might not be able to produce gaits with the varied behaviors needed to adapt to new environments or tasks, and limitations set by human engineers might discard many near-optimal areas of the search space.

Being able to find the right complexity balance when designing a controller can be very challenging. Any choice made early in the design process might not suit future use, and picking a single controller complexity for all different uses might end up being a costly compromise reducing performance significantly. We have experienced this challenge in our own work where experiments are performed with a four-legged mammal-inspired robot with self-modifying morphology in both simulation and hardware [2]. Balancing the need for a low complexity controller when evolving morphology and control in few evaluations in hardware without falling, and evolution in complex and dynamic environments requiring exotic ways of walking in simulations, has proven impossible with our earlier controller design [3].

In this paper, we introduce a new controller where the complexity can be set by a single parameter that addresses this limitation. We use a dynamic genotype-phenotype mapping, illustrated in Fig. 1, where higher complexity controllers map the genotypic space to a larger controller space than lower complexity controllers. This allows a more flexible gait either when an evaluation budget allows for longer evolutionary runs, or when the added flexibility is needed for coping with difficult environments. Less flexible gaits can be used when there is a stricter evaluation budget, for instance in real-world experiments. We have investigated the controller in simulation with our four-legged mammal-inspired robot, and found that different gait complexities are optimal under different evaluation budgets. We also verified this through initial tests on the physical robot in the real world. This suggests that our new controller concept will be useful for coping with the competing demands of freedom versus ease-of-learning, especially important when evolving on both virtual and real-world robots.

The contribution of this paper is as follows: We introduce the concept of a variable complexity gait controller, and show how this can be implemented for a quadruped robot. We then demonstrate its value through experiments in simulation, and verify the results with preliminary testing on a physical robot in the real world.

2 Background

Evolutionary robotics uses techniques from evolutionary computation to optimize the brain or body of a robot. It can be used directly to improve the performance of a robot, or to study

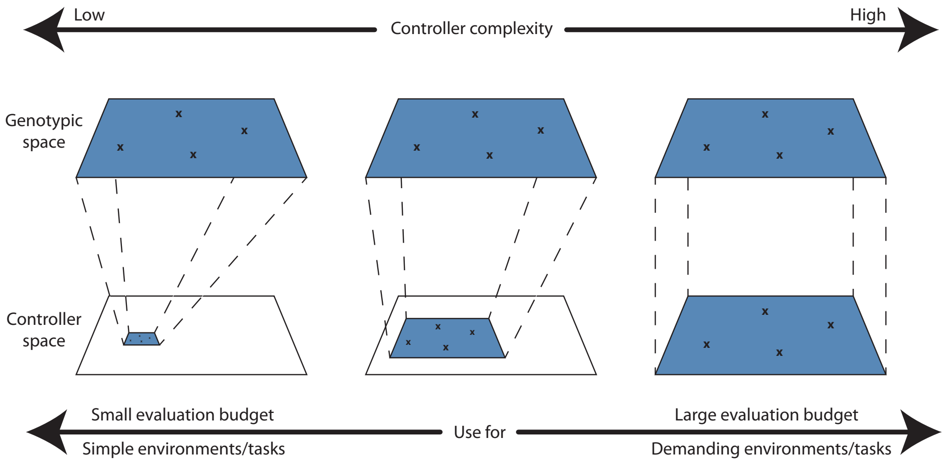


Figure 1: This diagram shows the concept of a variable complexity controller. The genotypic space is always the same size, but the mapping to controller space is changed by the controller complexity parameter, giving safer and more conservative gaits at lower controller complexities.

biological processes and mechanisms. When optimizing the brain of a robot, high-level tasks like foraging, goal homing or herding can be evolved, or lower level functions like sensory perception or new walking gaits. Optimizing the body of a robot allows adaptation to different tasks or environments, and research has shown that the complexity of evolved bodies mirror the complexity of the environments they were evolved in [4].

Several different types of optimization algorithms from evolutionary computation are used to optimize robot control. The most common is the Genetic Algorithm (GA) [5], which uses genetic operators like mutation and recombination to optimize gait parameters. It is often done using multiple objectives, in many cases achieving a range of solutions with different trade-offs in conflicting objectives, including speed and stability [6], or even speed, stability, and efficiency [7]. Evolutionary Strategies (ES) feature self-adaptation, by adding the mutation step size to the individuals. This has been shown to speed up the search, and in some cases outperform traditional EA approaches, when evolving quadrupedal robot gaits [8]. Genetic Programming (GP) represents individuals as tree structures rather than vectors, and has been shown to outperform simple GA algorithms when used to evolve quadruped gaits [9]. Quality-Diversity algorithms aim to build up an archive of solutions that exhibit different behaviors or characteristics that all perform as well as possible [10]. This set of diverse individuals then serves as a pool of solutions that can be searched through to find solutions to new problems, like a robot adapting to a broken leg [11].

Optimizing how a robot walks can be very difficult, and one of the biggest challenges is the bootstrap problem [1]. It can be very hard to start optimizing a robot gait if none of the random individuals tested initially provides a gradient towards good solutions. This is mostly a problem when optimizing in hardware, with much harder time constraints and potential physical damage to the robot. It can, however, also affect simulations, where initial individuals without any ability to solve a task can completely remove the selective pressure from the fitness functions needed for evolution to succeed.

There is a wide range of gait controller types used in evolutionary robotics, depending on what is being optimized. They are often divided into two categories, based on whether they work in the joint space, or Cartesian space [12]. A gait can either be represented as a few discrete poses with trajectories generated automatically between them, or as a continuous function that specifies the position or joint angles at all times. Some gait controllers use simple parameterized functions that control the joint space of the robot [11, 13]. Other gait controllers used in evolutionary experiments consist of a parameterized spline that defines each legs trajectory in Cartesian space. Evolution optimizes either the position of the spline points directly [8], or some higher level descriptors [6, 14]. Other controllers are based on central pattern generators of different architectures and models [15]. Some produce neural networks using techniques such as Compositional Pattern Producing Networks (CPPN), which has an inherent symmetry and coordination built-in. This can lead to gaits far surpassing the performance of hand-designed gaits based on parameterized functions [16].

The field of neuro-evolution often evolves the structure of the neural networks making up the gait controller, in addition to the connection weights. This goes against the general trend in other fields, where the complexity of gait controllers is most often kept static. Togelius defines four different categories [17]. *Monolithic evolution* uses a single-layered controller with a single fitness function. *Incremental evolution* in neuro-evolution has several fitness functions, but still one controller layer. *Modularised evolution* has more controller layers, but a single fitness function. *Layered evolution* uses both several controller layers, and several fitness functions. When evolving the complexity of a network, it has been shown that new nodes should be added with zero-weights [18], allowing evolution to gradually explore the added complexity.

3 Implementation

3.1 Robot

The experiments in this paper were performed on a simulated version of "DyRET", our four legged mammal-inspired robot with mechanical self-reconfiguration [3]. The robot platform is a fully certified open source hardware project, with source and details available online[1]. We use the Robot Operating System (ROS) framework for initialization and communication, and the simulated version runs on the Gazebo physics simulator. The robot and its simulated counterpart can be seen in Fig. 2.

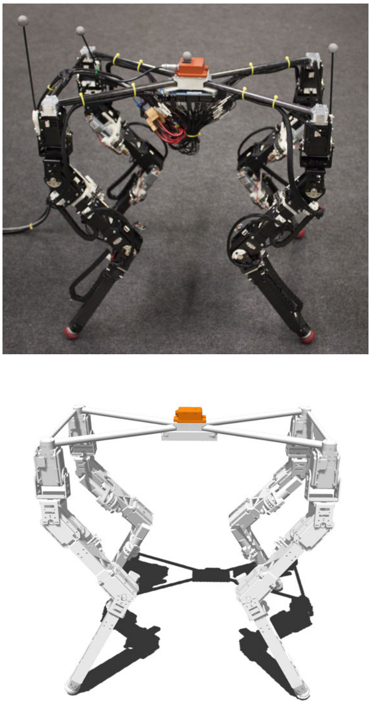


Figure 2: The physical robot on top, and the simulated robot on the bottom.

The robot uses Dynamixel MX-64 servos from Robotis in the hip joints, and Dynamixel MX-106 servos for the two lower joints. Its legs consist of two custom linear actuators each that allow reconfiguration of the leg lengths during operation. More mechanical details can be found in our previous work [3], and is not included here due to space constraints and the fact that we are mainly using a simulated version for our experiments.

3.2 Control

In our earlier experiments, we used a fairly standard parameterized spline-based gait controller working in Cartesian space. We have used the controller for evolving both control and morphology on the physical robot, with a complex search space with many degrees of freedom. This required us to have a low

[1] `https://github.com/dyret-robot/dyret_documentation`

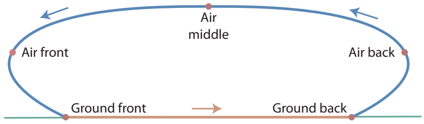


Figure 3: A simple example leg trajectory, seen from the side. The tip of the leg follows this path when the robot walks. The front of the robot is to the left.

complexity controller, but that meant it was not flexible enough to give us more complex gaits when we had higher evaluation budgets, such as when using simulations. Our goal was for the new controller to be adaptable to fit whatever needs we currently have or might have in the future, with a controller complexity that could be changed with a single parameter.

3.2.1 The gait controller

Since this gait is used on a physical mammal-inspired robot, the property of being learnable without excessive falling is important, and a much bigger challenge than for spider-inspired robots. We believe that a controller operating in joint space would not allow robust enough gaits at low controller complexity for our robot, so we chose to implement it in Cartesian space. There are many ways a gait can result in a fall, but ensuring that all legs on the ground are moving in the same direction with the same speed severely limits the chance of falling. Complementing this with a wide leg stance gives a good base to build a parameterizable gait controller on. Ensuring that only one leg is in the air at a time, and that the robot is always using the proper leg lift order, further helps the robot to remain stable.

3.2.2 Leg trajectory

The control system uses standard inverse kinematics to get the individual joint angles from the calculated positions. The leg trajectory is parameterized using an interpolating looping cubic Hermite spline, which intersects five control points. A simple example trajectory can be seen in Fig. 3. The start and end point of the spline are on the ground, while the other three points define how the leg moves forward through the air. The leg moves in a straight line on the ground, parallel to the body of the robot, so only two parameters decide their positions. The three points in the air are all three dimensional, with sideways movement being mirrored between left and right legs. This gives a total of 11 parameters that define the spline shape.

The two control points along the ground are sorted so that they always move the leg backward, while the order of the three control points in the air is chosen with an order resulting in the shortest possible spline. This ensures that no looping or self-intersection can happen, and allows all gait parameters to be set without constraints. A parameter for lift duration specifies the time the leg uses to lift back to the front, given in percentage of the gait period, while the frequency parameter gives the number of gait periods per second.

3.2.3 Balancing wag

In addition to positions generated for individual legs, a balancing wag is added to all legs. Due to the leg lift order, this can not be a simple circular motion, but needs different frequencies for the two axes. The movement allows the robot to lean away from the leg it is currently lifting, and gives better stability. Equation 1 shows how the wag is defined, with t defining the current time, and T the gait period. 0.43 is a factor to offset the movement between the two wag axes to align them with the gait. It has a phase offset (W_ϕ) that allows for tuning to dynamic effects of the robot, while amplitude can be set separately for the two directions (A_x/A_y).

$$\begin{aligned} W_x &= \frac{A_x}{2} * tanh(3 * sin(\frac{2\pi * (t + (W_\phi * T))}{T})) \\ W_y &= \frac{A_y}{2} * tanh(3 * sin(\frac{2\pi * (t + (W_\phi + 0.43) * \frac{T}{2})}{\frac{T}{2}})) \end{aligned} \tag{1}$$

3.2.4 Complexity scaling

The complexity of the controller can be modified by a single parameter, from 0 to 100%. There are many ways to provide a scaling of the complexity of the controller, but we chose to implement this using a dynamic genotype-phenotype mapping that varies the range of gait parameters linearly with the controller complexity. All controller parameters have a center value, that together with the minimum range gives the allowable range at controller complexity 0%. These have been chosen so they represent a very conservative and safe controller that should work well in most conditions, based on traditional robotics techniques and earlier experience with the robot. Using a more complex controller by allowing a large range of values, however, allows the controller to deviate from the safe values and into the more extreme values often needed for more complex environments or tasks. Parameters controlling the spline shape can be seen in Table 1, with high-level gait parameters in Table 2.

Table 1: Parameters and ranges defining the spline shape

Control point	Minimum	Maximum	Default value	Min range
Ground front	-150	150	50	50
Ground back	-150	150	-100	50
Air 1	[-25, -150, 10]	[25, 150, 80]	[0, 75, 30]	[0, 50, 10]
Air 2	[-25, -150, 10]	[25, 150, 80]	[0, 0, 50]	[0, 0, 10]
Air 3	[-25, -150, 10]	[25, 150, 80]	[0, -75, 50]	[0, 50, 10]

Table 2: Parameters and ranges of gait parameters.

Parameter	Minimum	Maximum	Default value	Min range
Wag phase	$-\pi/2$	$\pi/2$	0	0.2
Wag amplitudes	0	50	0	5
Lift duration	0.05	0.20	0.175	0.05
Frequency	0.25	1.5	-	-

Examples of splines with different gait complexities can be seen in Fig. 4. For complexities of 0, the splines are fairly conservative, but even though the parameter ranges are low, they do show some variation in their basic shapes. The higher complexity gaits have spline shapes that are much more unconventional, though sorting the control points to minimize spline length does remove self-intersections to keep all trajectories feasible. Please note that the plot shows the commanded position to the robot, and that the actual leg trajectory can be very different than commanded, due to the mechanical and control properties of the actuators, and the dynamics of the system. Very complex shapes that appear unintuitive for human engineers might end up giving much smoother and higher performing gaits in the real world than expected.

3.3 Evolutionary setup

Here we describe the setup we used for evolving the controllers, as well as how we evaluated them. We evolved controllers for both stable and fast forward walking on flat ground.

3.3.1 Evolutionary algorithm and operators

We used the NSGA-II evolutionary algorithm, running on the Sferes2 evolutionary framework. We chose this algorithm since we are optimizing both speed and stability, but would not like to choose the specific trade-off between the two objectives before optimization. NSGA-II features a mechanism to increase the crowding distance in the Pareto front, which gives a wide range of trade-offs to pick from.

Gaussian mutation was used with a mutation probability of 100% and a sigma of 1/6. No recombination operators were used.

Early experimentation showed a big difference in the number of evaluations before convergence for different controller complexities, which suggested the need for different population sizes. We tested a range of different population sizes at the minimum and maximum complexity, as well as a few points in between, and found that a population of eight at zero complexity, and 64 at full complexity worked best. Population sizes for all intermediary complexities were set linearly, and rounded to the nearest power of two. Tests showed that runs at all gait complexities converge to a satisfactory degree after 8192 evaluations.

We performed 25 runs for each controller complexity in simulations to gain a good estimate of the performance. Each simulated run took about 11 hours, and we used about 10,000 CPU core hours on the simulation for the experiments featured in the paper. Experiments in the real world take a lot longer, so we only performed three runs for each controller complexity, as the experiment only serves as a preliminary test to see confirm simulated results in the real world.

3.3.2 Fitness objectives

We used both speed and stability as our fitness measurements. Speed was calculated as the distance between start and stop position, divided by the evaluation time, as seen in equation 2. Distance was measured using motion capture equipment in the real world, and extracted directly in simulation. Only the speed straight forward was used, so we filtered out any sideways movement by only measuring position in the forward axis. Stability was calculated with a weighted sum of the variance in acceleration and orientation. The full fitness function for stability can be seen in equation 3, where *acc* are samples from the accelerometer, *ang* are samples from the orientation output of the Attitude and Heading Reference System (AHRS), i is the sample index, and j is the axis of the sample. The Xsens Mti-30 AHRS was used on the physical robot, and a virtual version of the same was used in simulation.

$$F_{speed} = \frac{\|P_{end} - P_{start}\|}{time_{end} - time_{start}} \tag{2}$$

$$G(A_j) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_{j,i} - \overline{A_j})^2}$$

$$F_{stability} = -\left(\alpha * \sum^{axes} G(Acc_{axis}) + \sum^{axes} G(Ang_{axis})\right) \tag{3}$$

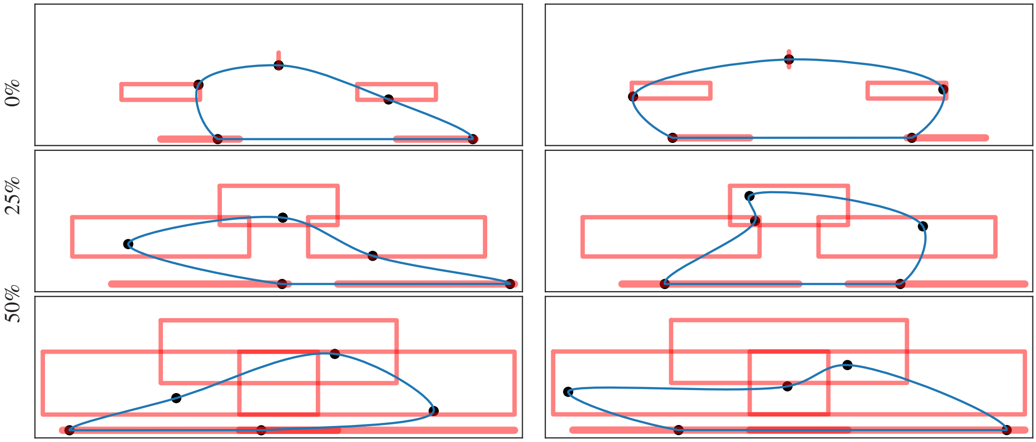


Figure 4: Examples of leg trajectory splines generated at different gait complexities. These are seen from the side of the robot, with the front of the robot to the left of the plot. The red boxes show the range of possible control point positions.

3.3.3 Evaluation

We ran all our simulations on the Gazebo physics simulator. Each gait was evaluated in simulation by walking forwards 1 meter, with a timeout of 10 seconds. The position and pose of the robot were reset between all evaluations.

Evaluating and comparing the performance of different optimization runs can be challenging when doing multi-objective optimization. This is especially true when using an algorithm like NSGA-II, that has a mechanism for stretching out the Pareto front, making it hard to compare the two objectives separately. Therefore, we instead looked at the hypervolume [19] when comparing populations. The hypervolume measures the volume (or area, in the case of two objectives) of the dominated part of the objective space. The lower bound of stability was set to -1 for the hypervolume calculation, while speed was capped to 0 m/min.

4 Experiments and results

We present the results of experiments in simulation and on a real-world robot. These experiments are simplified and performed with as many variables removed as possible. The robot's task is to walk straight forward, and the environment is a flat surface with medium friction, both in simulation and the real world.

4.1 Finding the maximum needed complexity

First, we wanted to investigate whether there is a maximum controller complexity needed for the environment and task we are using. Since neither is very challenging, we do not expect the need for very complex controllers. We ran full evolutionary runs at a range of gait complexities.

Fig. 5 shows how the hypervolume progresses over evaluations. This shows that the lower complexity controllers converge quicker, but are not able to achieve the same performance as the higher complexity controllers. The 50% and 100% complexity controllers end up with the same performance, though the 100% complexity controller takes considerably longer to converge.

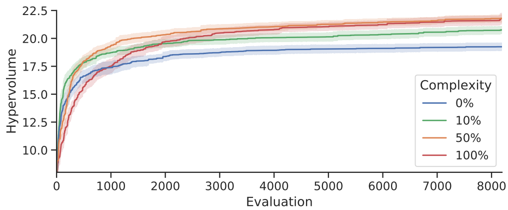


Figure 5: Hypervolume from evolutionary runs with selected gait complexities. The solid lines show the means, with 95% confidence interval in the shaded areas.

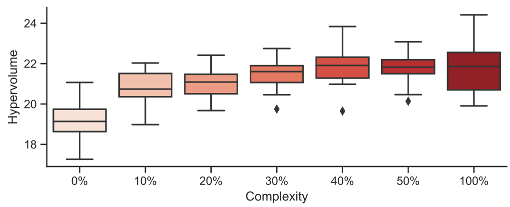


Figure 6: Hypervolume for runs with gait controller complexities ranging from 0% to 100%, showing details from the end results of the optimization process.

The details of the last evaluations of the runs are better illustrated with the boxplots, seen in Fig. 6. These show the distribution of the hypervolumes achieved at the end of all the optimization runs. The hypervolume improves for gait complexities from 0% to 40%, but there is no improvement between 40% and 50%. 100% complexity has a wider spread than the others, which might be beneficial in some applications, but the median performance is no better than the 40-50% complexity.

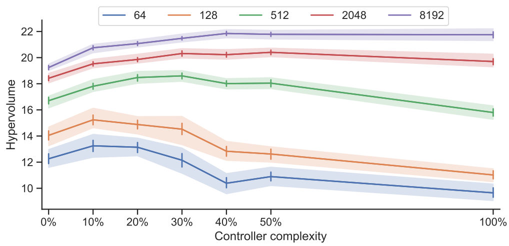


Figure 7: This figure shows how different controller complexities affects achievable hypervolume for different evaluation budgets. The vertical lines show the standard deviation, while the shaded areas show the 95% confidence intervals.

4.2 Complexity for different evaluation budgets

A potentially rewarding feature of controlling the complexity of the gait is the ability to adapt it to a specific evaluation budget. The price of computational resources is decreasing, enabling a large number of evaluations in simulation. Hardware experiments, however, are limited by the number of robots that can be built, maintained, and supervised during experiments. Evaluation is therefore much more expensive for hardware experiments than for simulations, and this gap will only increase.

For this investigation, we have selected a range of different evaluation budgets to test. We have previously used 64 and 128 evaluations in our hardware experiments [14, 2], and 512, 2048 and 8192 evaluations gives a range more appropriate for simulation experiments.

Fig. 7 shows how the controller complexity affects achieved hypervolume for the different budgets. For the shortest two simulation cases, with 64 and 128 evaluations, hypervolume is highest at 10% complexity. Budgets 512 and 2048 achieve the best performance around 30%, while the long simulation case performs best at 40%-100%.

4.3 Analyzing resulting populations

Figure 9 shows which parameters are tested at various parts of the search. Some parameters, like the *y* position of the back ground control point, end up close to their conservative estimate, and do not exploit their additional freedom from the higher complexity in our simple experiments, as seen in Figure 9a. Other parameters, like the *y* position of the front ground control point, do use more of their available range, although it is still close to its original estimate. In Figure 9c, the search with 50% controller complexity seems to maximize the *x* position of the third air control point in the spline, while with the whole area available in the 100% complexity controller, it ends up minimizing it.

4.4 Initial hardware testing

We also did evolutionary runs using this new controller on the physical robot in the real world with 64 evaluations per run, using eight generations of eight individuals. We decided to test a controller complexity of 0%, as well as 50%, which is the highest complexity we were confident in using on the physical robot without excessive risk of physical damage to the system.

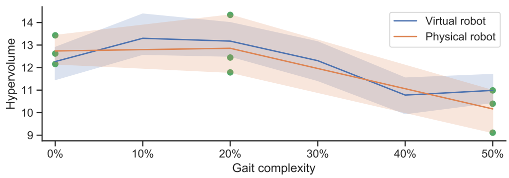


Figure 8: This figure shows the performance on the real robot, compared to the simulated version seen in Fig. 7. The added green dots are the resulting hypervolume from each of the runs in hardware.

We also tested 20%, which gives us another data point between these, and was among the two top performing complexities in simulation with this evaluation budget. The results can be seen in Fig. 8, where we can see the same general trends as in the simulator. Controller complexities 0% and 20% both did well, and we are not able to separate the two with the limited number of evaluations we were able to do in hardware. 50% controller complexity, however, does considerably worse than the other two, just as we saw in simulation.

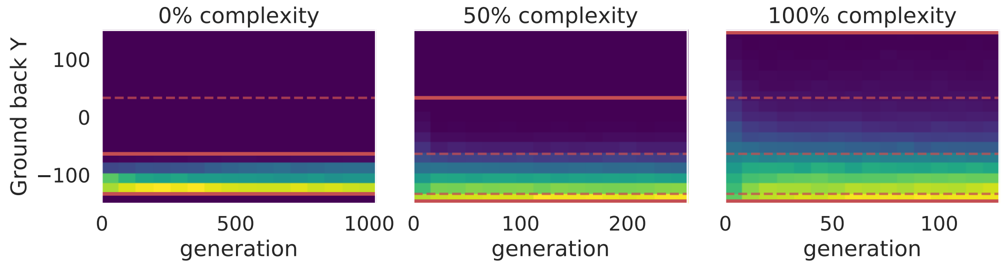
5 Discussion

The performance differences in Fig. 7 suggest that choosing the right controller complexity for an evaluation budget can be very important, especially when that budget is small. Lower complexity controllers fall less, so if optimization is done in hardware, this could also be taken into account when deciding on the complexity. We did a simple grid-search for our experiments since we were only investigating the controller, but more advanced search algorithms could be be performed to further optimise the choice of complexity.

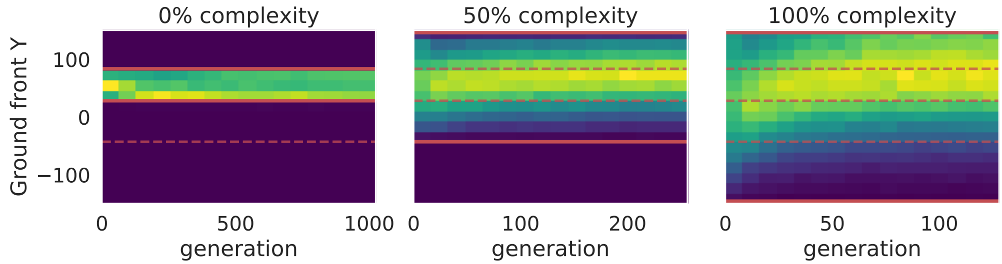
We used different population sizes when evolving with different complexities in our experiments. Our controller was designed to be evolved with evaluation budgets as small as 32 or 64 evaluations when doing real world experiments, and with budgets larger than 8192 when evolving in simulation. Limiting the population size to the smallest budget would give a very unrealistic measurement of performance for the larger budgets, and thus we chose suitable population sizes for the different complexities through simple trial and error. This does obfuscate the results to a degree, but we feel this gives the most fair comparison. The evolutionary operators would likely also be slightly different, but they were kept the same as they affect the search to a much smaller degree.

The parameter for the *x* position of the third air control point, seen in Fig. 9c, seems to be maximized at 50% complexity, but be minimized at 100% complexity. This is most likely due to interactions between different parameters. At half complexity, the optimal value might be towards the top of the parameter range. At full complexity, however, new ranges for the other parameters are opened up, allowing better performance for lower parts of the range.

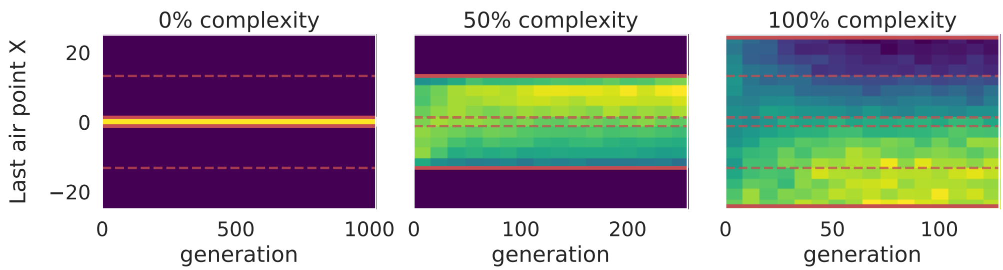
The choice of centers and minimum ranges for each gait parameter greatly affect the performance of lower complexity gait controllers. The choice should be based on conservative values that are assumed to work sufficiently in all environments, not on optimal values for a single environment. Evolution is often used to adapt to changes in environments or tasks. If the centers and ranges were chosen after optimal solutions were found, they would most likely not perform well when things change, and



(a) The y position of the second ground control point in the spline.



(b) The y position of the first ground control point in the spline.



(c) The x position of one of the air control points in the spline.

Figure 9: Values of a select few parameters throughout the optimization run. The solid red lines show the range of the parameters, and the dashed red lines trace the range from the other complexities to ease comparison.

one might as well just select the top performing individuals from simulation directly. In our case, we selected these values before doing the optimization, and several are far from optimal. This can be seen in Fig. 6, where the performance of the low complexity controller is much worse than for the higher complexity ones. This is by design, as safe and conservative parameters that work for all environments rarely do very well in any of them.

The choice of maximum ranges also affect the outcome, but not to as high a degree as the center and minimum range. Limiting the ranges too much means the controller will never be able to achieve the potential increase in performance from that specific controller feature. Having ranges that are too large, with values that will never be optimal under any circumstance, serves to slow down the search, and waste time and resources. A good optimization algorithm that is not getting stuck in early local optima, however, should be able to converge outside these infeasible areas. We therefore recommend anyone implementing this type of controller to spend some time choosing parameter centers and minimum ranges to be conservative and safe, but not be afraid to overshoot a bit on the maximum allowable range, as the consequence of choosing ranges that are too narrow is far worse than selecting too high.

Figure 8 shows the results from the testing in hardware. We are unable to say anything definitive with the results due to the low number of evaluations and the relatively high degree of noise, but it does support what we found in simulation. Not only did the 50% controller complexity perform worse, like predicted in simulation, but we also experienced qualitatively more extreme gaits, and actually had to pause the evolutionary runs at several times to repair the robot after damage. We also experienced several falls with the 50% controller complexity, but no falls or damage at the two lower complexities, supporting our original assumption that the gait values were conservative and safe.

We consider this type of controller to be very useful for researchers doing gait optimization in the real world on physical robots, as the reality gap can often times make it impractical or impossible to directly use individuals from simulation in the real world. Simulations can be used to find approximate upper bounds of the needed complexity as we saw in Fig. 6, but even more useful is being able to tune the complexity to the limited evaluation budget used in hardware, as seen in Fig. 7. We also expect that more demanding or dynamic environments and tasks might be able to exploit higher complexities better than what we experienced in our experiments, which only included forward walking in straight lines on even terrain.

6 Conclusion and future work

In this paper, we introduced our new gait controller with variable complexity. We tested the controller in simulation, and found that different gait complexities are optimal for different evaluation budgets. We also did preliminary tests on a physical robot in the real world that supported our findings. Being able to change the controller complexity allows a researcher to use less complex controllers when optimizing gait on a physical robot, and increase the complexity when needed for demanding environments, or when doing longer optimization in simulations.

One natural extension of our work is to use our variable complexity controller in incremental evolution. Since this controller offers a continuous complexity parameter, the difficulty can be gradually increased for each generation. Since an increase in difficulty follows a known set of rules, all individuals can keep their phenotypic values between generations, even when parameter ranges are expanded. This allows evolution to gradually explore the added complexity, in the same way that has been shown to be optimal for neuro-evolution [18]. The controller complexity can also be changed during the evolutionary process as part of evolutionary strategies, or be controlled during robot operation as part of lifelong learning.

We have only tested this controller in a single environment in simulation where complexities over 50% were not needed. It would be interesting to test it in more challenging and dynamic environments to see if controllers with higher complexities are able to use the increased parameter ranges to actually increase performance. Doing a more thorough investigation into the parameters selected might yield ranges or values that act limiting on the fully complex controller, and would allow even more flexible gaits. Analyzing the individual leg trajectories evolved would also be interesting, and could shed light on the matter from a different perspective. Investigating how evolutionary meta-parameters interact with the complexity would be interesting, including population size and evolutionary operators. Adding sensing and allowing the robot to choose which complexity is needed for its current environment is also worth exploring.

Acknowledgement

This work is partially supported by The Research Council of Norway under grant agreement 240862.

References

[1] Mouret, J., Doncieux, S.: Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In: 2009 IEEE Congress on Evolutionary Computation. pp. 1161–1168 (May 2009)

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

[3] Nygaard, T.F., Martin, C.P., Torresen, J., Glette, K.: Self-modifying morphology experiments with dyret: Dynamic robot for embodied testing. In: 2019 IEEE International Conference on Robotics and Automation (ICRA) (May 2019)

[4] Auerbach, J.E., Bongard, J.C.: Environmental influence on the evolution of morphological complexity in machines. PLoS computational biology 10(1) (2014)

[5] Gong, D., Yan, J., Zuo, G.: A review of gait optimization based on evolutionary computation. Applied Computational Intelligence and Soft Computing (2010)

[6] Golubovic, D., Hu, H.: Ga-based gait generation of sony quadruped robots. In: Proc. 3th IASTED Int. Conf. Artificial Intelligence and Applications (AIA) (2003)

[7] Moore, J.M., McKinley, P.K.: A comparison of multiobjective algorithms in evolving quadrupedal gaits. In: International Conference on Simulation of Adaptive Behavior. pp. 157–169. Springer (2016)

[8] Hebbel, M., Nistico, W., Fisseler, D.: Learning in a high dimensional space: Fast omnidirectional quadrupedal locomotion. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X (2007)

[9] Seo, K., Hyun, S., Goodman, E.D.: Genetic programming-based automatic gait generation in joint space for a quadruped robot. Advanced Robotics 24(15) (2010)

[10] Pugh, J.K., Soros, L.B., Szerlip, P.A., Stanley, K.O.: Confronting the challenge of quality diversity. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 967–974. ACM (2015)

[11] Cully, A., Clune, J., Tarapore, D., Mouret, J.B.: Robots that can adapt like animals. Nature 521(7553), 503 (2015)

[12] de Santos, P.G., Garcia, E., Estremera, J.: Quadrupedal locomotion: an introduction to the control of four-legged robots. Springer Science & Business Media (2007)

[13] Nygaard, T.F., Samuelsen, E., Glette, K.: Overcoming initial convergence in multi-objective evolution of robot control and morphology using a two-phase approach. In: Squillero, G., Sim, K. (eds.) Applications of Evolutionary Computation. pp. 825–836. Springer International Publishing (2017)

[14] Nygaard, T.F., Torresen, J., Glette, K.: 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)

[15] Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. Neural networks 21(4), 642–653 (2008)

[16] Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J.C., Lipson, H.: Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In: ECAL. pp. 890–897 (2011)

[17] Togelius, J.: Evolution of a subsumption architecture neurocontroller. J. Intell. Fuzzy Syst. 15(1), 15–20 (Jan 2004)

[18] Tomko, N., Harvey, I.: Do not disturb: Recommendations for incremental evolution. In: Proceedings of ALIFE XII, the 12th International Conference on the Synthesis and Simulation of Living Systems (2010)

[19] Knowles, J.D., Corne, D.W., Fleischer, M.: Bounded archiving using the lebesgue measure. In: The 2003 Congress on Evolutionary Computation. vol. 4 (Dec 2003)

Paper V

Environmental Adaptation of Robot Morphology and Control through Real-world Evolution

Tønnes F. Nygaard, Charles P. Martin, Jim Torresen, Kyrre Glette

Under review

V

Paper VI

A Morphologically Adaptive Quadruped Robot in the Wild

Tønnes F. Nygaard, Kyrre Glette, Charles P. Martin, Jim Torresen, David Howard

Under review

This paper was originally submitted in a single-column format, but has been revised to fit in the two-column thesis template.