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.**

I. 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

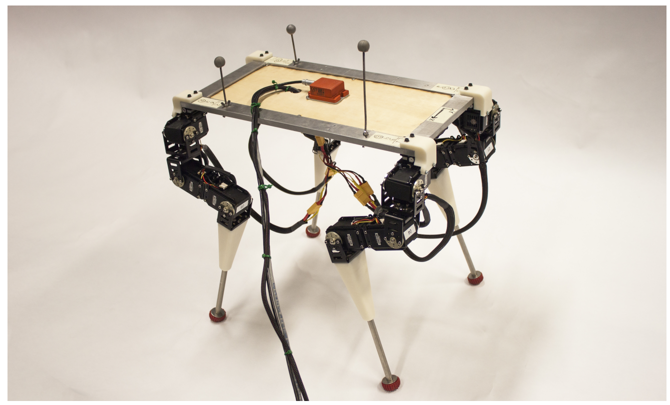


Fig. 1: The robot used for the experiments.

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.

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.

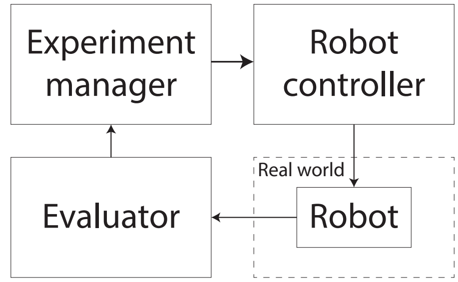


Fig. 2: An overview of the system.

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

II. Robot and evolutionary setup

An overview of the experimental system can be seen in Fig. 2. An experiment manager controls the system and the evolutionary algorithm. The robot controller receives commands from the experiment manager and controls the physical robot. An evaluator calculates the performance of the robot, which is then sent back to the experiment manager for use in the fitness functions.

A. 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 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.

B. 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. 3. 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

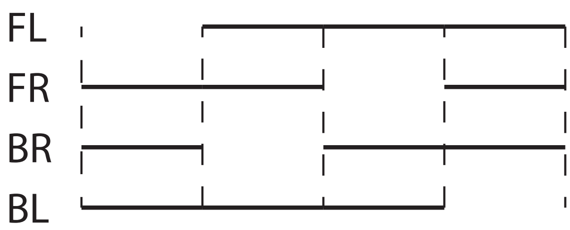


Fig. 3: The gait sequence of the robot. Solid lines mark ground contact, and leg positions are given according to front (F), back (B), left (L) and right (R).

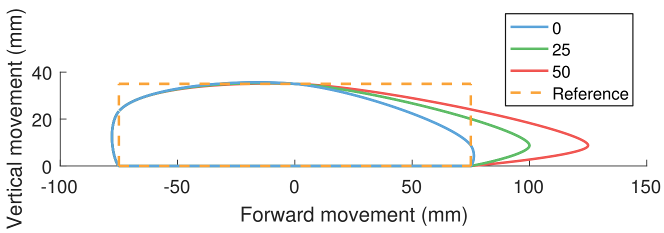


Fig. 4: The resulting spline with *step_length* 150, *step_height* 35, and three different *step_smoothing* parameters. A reference rectangle with the chosen *step_height* and *step_length* is shown for comparison.

for its interpolating nature and relative resistance to self-intersection.

The gait generator use parameter ranges defined in Table I 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 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 horizontal direction. Examples of a typical spline with different *step_smoothing* parameters can be seen in Fig. 4.

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.

[1]Details on control point generation can be found in the source code available for download at http://robotikk.net/papers/ICES16

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 I: 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 II: Parameters for the evolutionary setup

The control system is implemented in C++ using the kinetic distribution of 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.

C. 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 I when testing a candidate. Initial tests showed that gaussian mutation on all genes with a sigma of 1/6 and no recombination worked well. The evolutionary parameters can be seen in table II.

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.

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

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

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

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

D. 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.

III. Experiments and Results

A. 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.

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

TABLE III: Parameters for the different evolutionary runs.

Parameters for the different runs are given in table III. 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 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.

B. 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[2].

The SO optimization of speed resulted in the fitnesses seen in Fig. 5. 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. 6 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.

The fitness from the SO optimization of stability can be seen in Fig. 7. This figure shows convergence after only 4 generations, though with a slight increase in fitness in the last three generations. Fig. 8 shows which parameters are tested throughout the run, and we can see that *step_length* is centered

[2]Videos of re-evaluations can be seen at http://robotikk.net/papers/ICES16

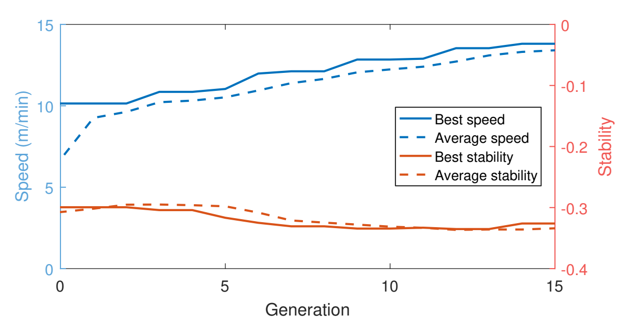
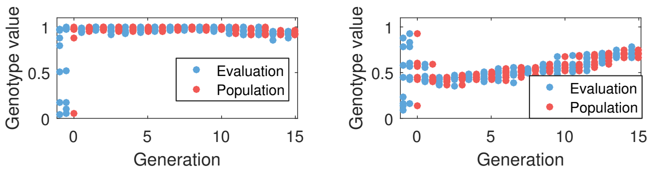


Fig. 5: Fitness results from speed optimization. Stability is not used for fitness evaluation and only included for visualization.



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

Fig. 6: Parameters in the SO optimization of speed. The changing population is seen, as well as the evaluated parameters between each generation.

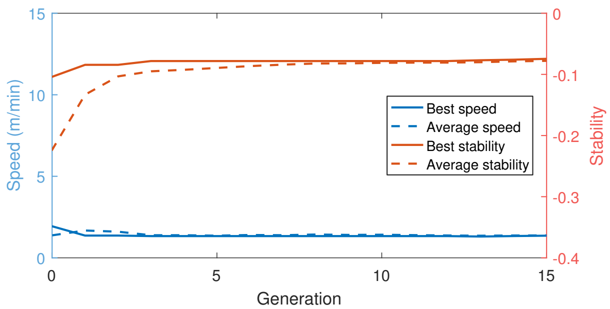
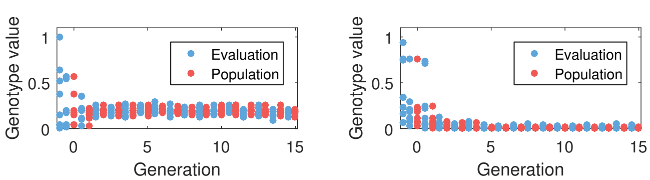


Fig. 7: Fitness results from stability optimization. Speed is not used for fitness evaluation and only included for visualization.

around 20% throughout the run, while the *gait_frequency* is quickly minimized.

Fig. 9 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.

Fig. 10 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



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

Fig. 8: Parameters in the SO optimization of stability. The changing population is seen, as well as the evaluated parameters between each generation.

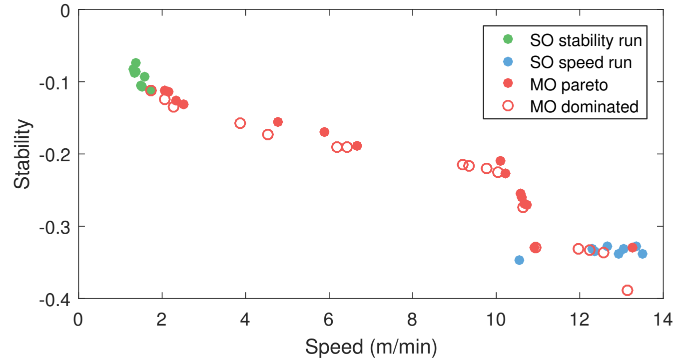
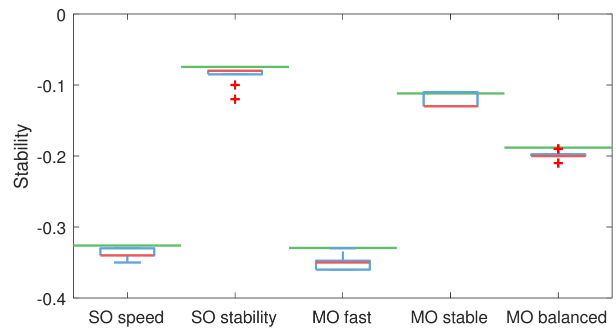
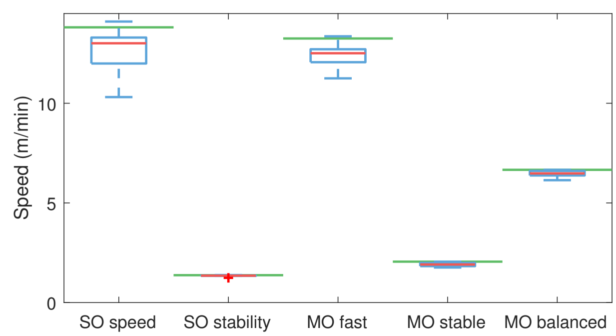


Fig. 9: Pareto front from the MO run, along with individuals from the SO runs of speed and stability.



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



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

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

one individual from the middle of the Pareto front. We see that the measurement noise 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. 11 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.

IV. 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. 9, 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. 5, 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. 7 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. 11 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. 11 that the most stable solution with stability in Fig. 11f also has one of the least constant speeds, seen in the distance walked in Fig. 11e. 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. 10 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 I, 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 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.

V. 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 sta-

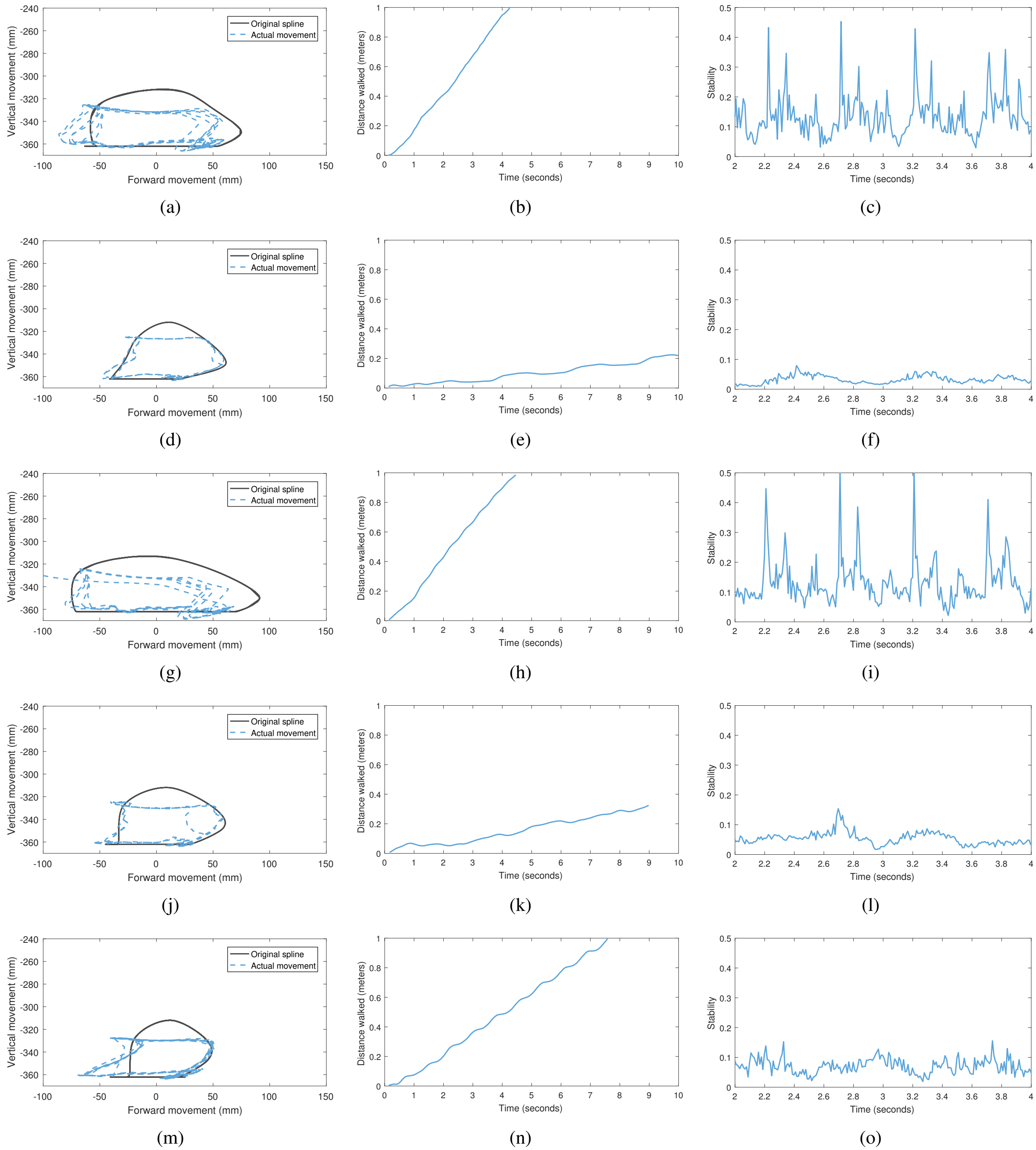


Fig. 11: First column is the gait spline of the individual, second column is the distance from the starting position, and the third column is the stability of the gait. Subfigure (a) to (c) is the fastest individual from the SO optimization of speed, (d) to (f) is the most stable from the SO optimization 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.

bility 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.