Concurrent Optimization of
Mechanical Design and Locomotion Control of a Legged Robot*

K. M. DIGUMARTI[†], C. GEHRING[†/‡], S. COROS[‡],

J. HWANGBO[†] and R. SIEGWART[†]

[†]*Autonomous Systems Lab, ETH Zurich,* [‡]*Disney Research Zurich*
8092 Zurich, Switzerland, gehrinch@ethz.ch

This paper introduces a method to simultaneously optimize design and control parameters for legged robots to improve the performance of locomotion based tasks. The morphology of a quadrupedal robot was optimized for a trotting and bounding gait to achieve a certain speed while tuning the control parameters of a robust locomotion controller at the same time. The results of the optimization show that a change of the structure of the robot can help increase its admissable top speed while using the same actuation units.

Keywords: Legged Robots; Optimization; Locomotion Control; Morphology

1. Introduction

Progress in the development of legged robots, as epitomized by Honda's Asimo[1] or Boston Dynamics' BigDog,[2] make legged robots increasingly more attractive for real-world applications. Designing a legged machine for a specific application is a challenging optimization problem because it needs to fulfill multiple constraints at once. On one side, the very nature of legged locomotion introduces large stress in the mechanical structure resulting from high impacts. On the other side, limited performance of available components such as actuators and energy sources poses further challenges. The morphology of a legged robot is therefore commonly determined by the available hardware and such that issues arising from stress, packaging and other engineering problems can be solved. During the design process only very rough calculations of simple tasks like jumping are often considered. The control aspect of the complex task is mostly taken care of after the robot is built. The controllers are consequently constrained to work within

limitations imposed by the hardware. This is in contrast to nature's evolutionary optimization, through natural selection, which takes both morphology and locomotion characteristics into account to achieve an optimal fit between design and ecology.[3] The influence of morphology on performance and the cross-coupling between spatio-temporal gait characteristics (stride lengths and frequencies, step lengths, duty factors, etc.) and the structure is very complex and to the best of our knowledge, there are no analytic or simplified models that can capture it succinctly. Our approach therefore is to simultaneously optimize the design and control of the robot using realistic and sufficiently detailed models of the robot, engineering constraints as well as the impact of morphology on the performance of the robot.

A recent publication about actuator sizing[4] highlights the need for such design rules. Khan et al. compute, for two simple tasks, a map that relates overall weight to segment length, and peak joint torques and velocities. The resulting maps indicate the requirements for the actuators to achieve a jumping task or trotting at a certain speed. The drawback of such simple methods is that they do not scale well with the parameter space that is actually required to design a robot. Karl Sims[5] was one of the first that simulated complex 3D worlds, where creatures had to compete against each other in an arena. The characters evolved by altering the morphology including number of bodies, type of joints, as well as actuators and sensors. Neural systems were used as locomotion controllers that could adapt to the morphology and vice-versa. A genetic algorithm generated many different successful creatures with different behaviors. Lipson et al.[6] showed how useful such an optimization can be if it is tailored to real robots. Larpin et al.[7] demonstrated how co-evolution of morphology and control can be used to develop steerability in virtual quadrupeds. They claim that a robot morphologically optimized specifically for forward motion or for energy efficient motion cannot obtain the new capability just by evolving control parameters.

In previous work,[8] we optimized the parameters of a robust locomotion controller[9] for an electrically-driven medium-dog-sized quadrupedal robot called StarlETH.[10] The optimization approach allowed us to find control parameters for two agile gaits, the pronk and the bound, which were too difficult to find manually [a]. The solutions found in simulation could be applied directly to the real robot, because many of the hardware limitations

---

[a]Video of StarlETH performing a running trot, pronk and bound:
`http://youtu.be/Tj1wreifYhU`

(a) Quadruped StarlETH

(b) Gait pattern

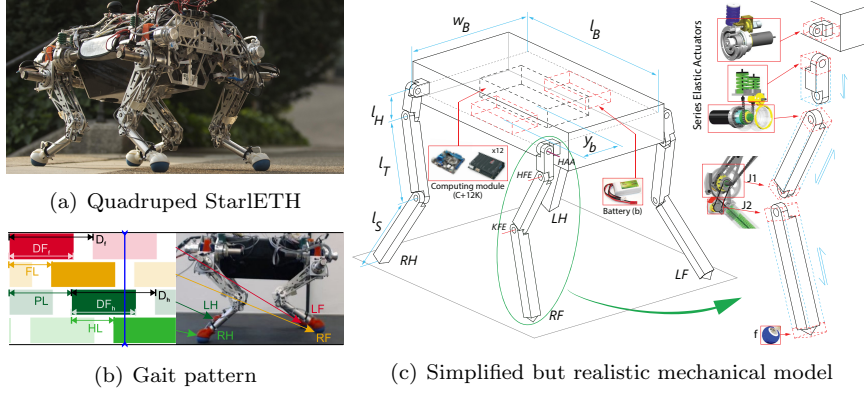(c) Simplified but realistic mechanical model

Fig. 1.   Parameterization of the structure and the gait pattern as part of the controller

were considered in the optimization. By using the same framework, we add kinematic and dynamic parameters to the optimization task and try to optimize different gaits for different walking speeds, aiming, in particular for top speed. The results of the concurrent optimization are discussed and compared to the current design of the robot to generate design-guidelines for a future version of the robot. We reduce the high-dimensional parameter search space that is imposed by the complexity of the design problem by posing specific design questions. In particular, we look for a similarly sized quadruped that uses the same actuation units, but can achieve higher running speeds.

2. Parameterization of the Design and Control of the Robot

The quadrupedal robot StarlETH[10] shown in Fig. 1(a) has four articulated legs, each with three degrees of freedom namely hip abduction/adduction (HAA), hip flexion/extension (HFE) and knee flexion/extension (KFE). The joints are driven by series elastic actuators that enable torque control of the joints.

Our objective is to optimize the running speed for pre-defined dynamic gaits. The actuators are to be re-used in the new design. We wish to keep the robot symmetric to let it run both forwards and backwards with equal capability, i.e. we search for four identical leg designs. The configuration of the legs, i.e. whether the knees on one side face each other, is not part of the search space either.

2.1. *Design Parameters*

A simulation framework based on the Open Dynamics Engine[11] is used to simulate the multibody dynamics of the quadrupedal robot. We model the spherical feet as point contacts and neglect the dynamics of the series-elastic actuators. The desired joint torques during the stance phase of the legs are thus directly used in the simulation, whereas a simple PD-law is used to compute the torques from the desired joint positions during the swing phase of the legs. The gains of the actuator controllers were not tuned in the previous work, but need to be considered in the optimization when the inertial parameters of the legs change.

Fig. 1(c) shows a simplified model of the robot. The components of the robot are separated into two categories based on their mass and inertia contribution. Mechanical parts such as actuators, springs and major bearings that have a significant mass and moment of inertia and those that we wish to retain irrespective of the design are considered to be fixed. These are shown in Fig. 1(c) with a red dotted outline. Other components that can have a varying mass and inertia are shown with a blue outline. The dimensions of these components are parameters that the optimizer is free to determine. The mass and inertia of these parts are computed online and the centres of mass are adjusted accordingly.

The main body(B) is composed of the basic structure(F), an on-board computer(C), motor controllers(K) for each of the joints, battery packs(b), a collection of sensors and additional cabling distributed throughout the structure. The simplified model assumes that the computer and the motor controllers are placed in the centre of the frame due to packaging problems and that the frame also includes weight of the cabling. The position of the battery packs within the frame is parameterized. In order to keep the robot symmetric and allow it to walk both forwards and backwards with equal ability, we restrict the batteries to lie on the lateral axis. At the juncture of the body with each hip, is a mass resulting from the HAA actuator and corresponding elastic elements.

Each leg of the robot consists of three segments: the hip(H), the thigh(T) and the shank(S). The hip consists of a parameterized length element(h) and a fixed mass element consisting of elastic elements(E) of the HFE series elastic actuator. The thigh has a central element(t) of parameterized dimensions and fixed mass elements at either ends. The one closer to the hip is made up of a pair of actuators (HFE, KFE), gears and a few significant parts. The other end is composed of parts of the encoder and connecting mechanism(J1). The shank is parameterized in a similar fashion. Its upper

end is largely made up of the connecting mechanism(J2) while the lower extreme is equipped with a foot(f) of fixed mass. Five of the six design parameters affect the mass of the robot as follows:

$$m_B = m_C + 12m_K + 2m_b + 4m_{HAA} + m_F(l_B, w_B)$$
$$m_H = m_E + m_h(l_H)$$
$$m_T = m_{HFE} + m_{KFE} + m_{J1} + m_t(l_T)$$
$$m_S = m_{J2} + m_f + m_s(l_S).$$

Inertia of each of the above mechanical components is computed in a similar fashion, taking into account the spatial distribution of mass elements and locations of their centers of mass. This simplification may not fully account for mechanical stress, but should nevertheless generate meaningful results.

2.2. *Control Parameters*

The desired motion of the trunk such as the height above ground and pitching orientation, as well as the vertical motion of the feet during the swing phase are defined by parameterized trajectories that are optimized.[8] The foothold locations are regulated using simple controllers based on an inverted pendulum that keep the robot balanced. The limb coordination and the timings of the swing and stance phases of the legs are parameterized based on the anteroposterior sequence of movement (APS) theory proposed by Abourachid et al.[12] This parameterization, as shown in Fig. 1(b) can represent both symmetrical and asymmetrical gaits, enabling us to predefine the type of gait, and gives a reduced number of variables that is preferable for the optimization. The generated desired motion is tracked using a virtual model control method in combination with a force distribution that keeps the desired contact forces within the friction cones. Some of the parameters of the motion control are also determined by the optimizer. More information on the locomotion controller is given in [9] and more details about the parameterization is found in.[8]

3. Optimization

To solve the high-dimensional, non-linear and non-smooth optimization problem we employ a direct policy search method based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES).[13] We evaluate the performance of the robot by means of a cost function that we seek to minimize. In a generic form, the optimization problem can be formulated as $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum w_i \cdot C_i(\boldsymbol{\theta}, \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau})$, where the objective is to find an op-

timal parameter set $\boldsymbol{\theta}^*$ by minimizing a weighted sum of costs C_i, which depend on the state trajectory of the robot $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and the actuation signals $\boldsymbol{\tau}$. A variety of cost functions can be used to guide the parameter set towards an optimum. For instance, we ensure that the resulting motions are smooth by penalizing changes in torque between successive control steps. We impose a cost on collisions between limbs so that the solution is natural and physically plausible. We also seek a solution that is power efficient while ensuring that the peak power in each joint is well within conservative limits of the actuator (150 W). Similarly we penalize joint velocities as soon as they exceed 80% of motor limits (8 rad/s). Joint torques are clamped to the maximum permissible value (20 Nm) before applying to the robot in simulation. We also experimented with various criteria to penalize falls. As a simple measure, we ensure that all the joints are within respective position limits which limit the range of motion and thus implicitly the performance. Additionally we tried penalizing the robot for stumbling by imposing a high cost when parts other than the feet touch the ground. However, we noticed that this impedes the motion of the robot and it prefers taking little steps to large ones to reduce the risk of failure.

Each parameter set receives a score based on the performance of the robot in the given task, which is, to accelerate to a peak velocity and maintain it. The robot initially starts at rest. It slowly accelerates until it reaches the desired maximum speed within a span of 15 strides. It then maintains that speed for a further 20 strides. Using a steeper profile for acceleration is not advisable as it introduces a sudden jerk in the robot causing it to destabilize. Ignoring the period of constant velocity might result in solutions that can only reach the maximum velocity but not walk stably at that velocity.

Since the locomotion controller depends on a model of the robot, modeling errors may affect the performance of the robot. While this is always a concern, given that we have been able to use parameters optimized in simulation on our robot in the past,[8] we believe the results of the simulations are trustworthy. What we found very important is to consider the effect of sensor noise. More important is to consider the estimator that is required to provide the full state of the robot to the controller. We therefore increase the difficulty after each optimization batch from ideal state information, to estimated states using an extended Kalman filter[14] and further up to simulated sensor noise.
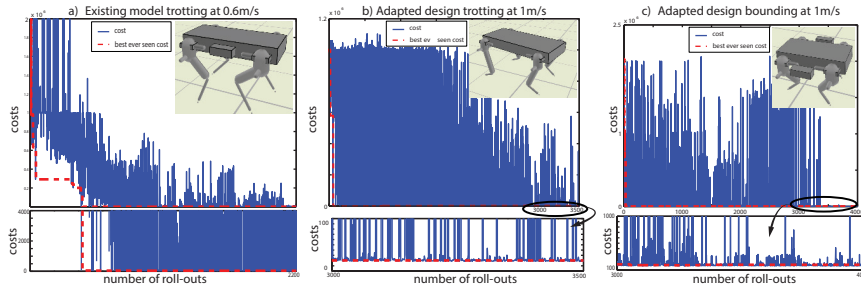
Fig. 2. Cost convergence in optimized model: Figures (a), (b) and (c) show convergence of cost during optimization. Best seen cost is shown in red. Inset: models representing existing configuration (a), mechanical design optimized for trotting at 1m/s (b) and mechanical design optimized for bounding at 1m/s (c).

4. Results

As a first objective we wanted to determine the maximum heading velocity under a trotting gait that the existing StarlETH configuration could achieve. With the current configuration and a relatively low speed of 0.25m/s we optimized for the controller parameters and gait parameters.

We fixed a minimum stride duration of 0.6s for the trot for cosmetic reasons. Although experiments with smaller stride duration give results that are feasible, the robot has a more crouched posture as against a desirable upright posture. We let the robot choose between a walking and a flying trot. The optimized parameter set was then used as a starting point for the next task with an increased maximum desired speed.

Stable solutions were found for speeds up to 0.6m/s. Beyond this speed, the solutions did not converge. We deduce that speeds in excess of 0.6m/s probably lie outside the feasible region of the robot i.e. they cannot be achieved with the current robot configuration. In previous experiments with the robot, we found very similar speed limits.

As a second objective we let the robot simultaneously optimize its design, under identical task settings, to find out if its top speed could be improved. We found out that co-evolving the design could easily push its maximum speed up to 1m/s. It is of value to note that the same power and torque limits of the actual robot were imposed even when optimizing the design. Figure 4 shows convergence of costs over the course of the optimization. This took about 1000 roll-outs for the existing configuration and about 3500 roll-outs while adapting mechanical design. A total of 31 parameters were optimized of which 6 dictate the design. One interesting observation

Table 1.   Comparision of Existing Design and Optimized Design.

	existing design		%change for optimized design	
	trot	bound	trot	bound
top speed [m/s]	0.6	0.6	+67	+67
stride duration [s]	0.6	0.29	0	-31
duty factor [% of stride dur.]	49	43	+2	+16
pair lag [% of stride dur.]	-	41	-	-25
body length [m]		0.505	+38	-26
body width [m]		0.37	+21	-8
body mass [kg]		17.16	+7	-3
hip length [m]		0.0685	0	0
hip mass [kg]		0.38	0	0
thigh length [m]		0.2	+50	-25
thigh mass [kg]		1.87	+23	-11
shank length [m]		0.235	-36	-36
shank mass [kg]		0.32	-9	-9
position of battery [m]		0.185	-100	+35

is that the robot chose a duty factor of 0.5 which means that it prefers a walking trot to a flying one. Table 1 shows a comparison of the adapted design against the existing configuration. There is a significant increase in the size of the body. The robot also prefers to have its batteries closer to the centre. There is a noticeable change in the moment of inertia (+50% about sagittal axis, +96% about vertical axis and +85% about lateral axis). The thigh is longer than on the existing model while the shank is shorter. The results can be best seen in the accompanying video[b].

Similar experiments were done using a bounding gait. The maximum speed achieved by the existing configuration was 0.6m/s and once again we were able to push it to 1m/s by adapting the design. Here we optimized for 27 gait and control parameters and 6 design parameters. It is evident that the robot prefers to have a smaller profile as compared to the previous case. There is a significant reduction in the overall size of the body as well as the length of the leg. Mass distribution in the body is different as the batteries, in this design, are placed farther away from the centre.

5. Conclusion

We have demonstrated that concurrently optimizing the mechanical design and control parameters for quadrupedal robots can significantly improve

---

[b]http://youtu.be/Zac4iR6SZys

their motor capabilities. Hence, we want to investigate the found solutions in more detail. It is of interest to us to use this work as a guideline and test designs on robot hardware we plan to build in the future. There is scope for improving the accuracy of modeling by including details such as material constraints and mechanical failure to further facilitate practical applicability of predicted designs. Another aspect we are focusing on is to apply this tool to several different gaits like pace and gallop. Lastly, we are interested in finding morphologies that adapt to a combination of locomotion tasks such as mixture of gait patterns or tasks on unstructured terrain.

References

1. Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki and K. Fujimura, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2002).
2. M. Raibert, Bigdog, the rough-terrain quadruped robot, in *Proceedings of the 17th IFAC World Congress*, ed. M. J. Chung, (1)2008.
3. a. Herrel, a. Zaaf, B. Vanhooydonck, P. Aerts and R. Van Damme, *Netherlands Journal of Zoology* **50**, 261(January 2000).
4. H. Khan, C. Semini, D. Caldwell and V. Barasuol, Actuator sizing for highly-dynamic quadruped robots based on squat jumps and running trots *World Scientific* (World Scientific, August 2013).
5. K. Sims, Evolving 3D Morphology and Behavior by Competition *Artificial Life* **1**January 1994.
6. H. Lipson and J. B. Pollack, *Nature* **406**, 974(August 2000).
7. K. Larpin, S. Pouya, J. van den Kieboom and A. J. Ijspeert, *2011 IEEE Int. Conf. on Robotics and Biomimetics* , 2799(December 2011).
8. C. Gehring, S. Coros, M. Hutter, M. Bloesch, H. M. A. and R. Siegwart, Towards Automatic Discovery of Agile Gaits for Quadrupedal Robots, in *IEEE Int. Conf. on Robotics and Automation*, 2014.
9. C. Gehring, S. Coros, M. Hutter, M. Bloesch, H. M. A. and R. Siegwart, Control of Dynamic Gaits for a Quadrupedal Robot, in *IEEE Int. Conf. on Robotics and Automation*, 2013.
10. M. Hutter, C. Gehring, M. Bloesch, M. Hoepflinger, C. Remy and R. Siegwart, StarlETH: a compliant quadrupedal robot for fast, efficient, and versatile locomotion, in *Proc. of the Int, Conf. on CLAWAR*, 2012.
11. R. Smith, Open dynamics engine v0. 5 user guide *Page (s)* 2006.
12. A. Abourachid, *Comptes Rendus Biologies* **326**, 625(July 2003).
13. N. Hansen and S. Kern, *Parallel Problem Solving from Nature-PPSN* (2004).
14. M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy and R. Siegwart, *Proceedings of Robotics: Science and Systems* (2012).