IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Vitruvio: An Open-source Leg Design Optimization Toolbox for Walking Robots

Michael Chadwick, Hendrik Kolvenbach, Fabio Dubois, Hong Fai Lau, and Marco Hutter

Abstract— **We present an open-source framework for developing optimal leg designs for walking robots. The leg design parameters (e.g. link lengths, transmission ratios, and spring parameters) are optimized for a user-defined metric such as the minimization of energy consumption or actuator peak torque, enabling the user to better navigate through the high-dimensional and unintuitive design space. Our approach uses the single rigid body dynamics trajectory optimization tool *TOWR* to generate realistic motion plans. The planned predefined forces and motions are then used to identify actuator velocities and torques. Next, the leg design parameters are optimized using a genetic algorithm. The framework was validated by comparison with measured data on the *ANYmal* quadruped robot for a trotting motion, with errors in cumulative joint torque and mechanical energy each below 8% per gait cycle. Optimization of the *ANYmal* link lengths demonstrate that reductions in joint torque, mechanical energy, and mechanical cost of transport in the range of 5-10% are attainable.**

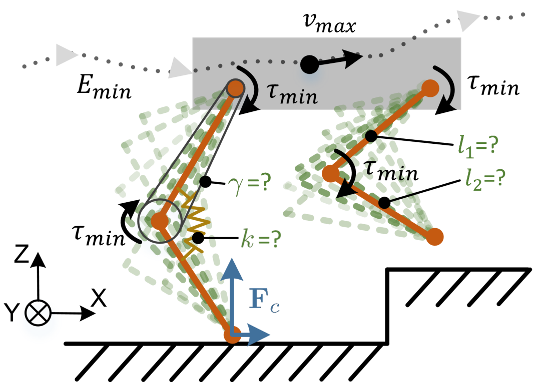

Fig. 1: A graphical representation of the design optimization problem. Given a motion task and a robot model, the framework returns an optimized leg design.

I. INTRODUCTION

Developing legs for walking robots is a daunting task due to the numerous design choices that significantly influence their performance. The design space of these robots is generally constrained by application-driven system-level requirements (e.g. walking speed, operating time, payload) and the choice of actuators. In this regard, the usage of the leg's linkage system may promote high performance because it allows the continuous selection of link lengths and the integration of transmission or elasticities (Figure 1). However, it compounds the design problem by increasing the number of possible designs with ambiguous trade-offs. Hence, manually finding the optimal solution is virtually intractable.

In order to circumvent this complicated design space, some designers took inspiration from nature. Examples include MIT Cheetah [1], [2], Cheetah-Cub [3], StarlETH [4], *ANYmal* [5], HyQ [6], and work based on Gravitationally Decoupled Actuation [7]. While this biomimetic approach leads to feasible leg designs, it may not be optimal for a given robot and task because of the fundamental structural difference between robots and animals. For example, when rotary actuators are used, pantograph legs tend to be energetically inefficient [8], and reptilian or insectile legs require unnecessarily high joint torques in stance [9]. Moreover, some tasks for which a robot should be designed for, such as heavy lifting and load carrying, may not exist naturally.

Computational optimization methods offer an alternative approach for finding solutions in such a design space. Here, the engineering problem is modelled mathematically, and the optimal solution is obtained numerically [10]. Since Sims' *Evolving Virtual Creatures* [11], genetic algorithms have been central to design optimization. An evolutionary strategy based on Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) [12] was used in [13] to simultaneously adapt the morphology and controller parameters to increase the top speed of a quadrupedal robot, subjected to actuator limits. Other CMA-ES approaches for the optimization of design morphology, control parameters, and gaits were presented in [14]–[16].

While these approaches obtain optimal solutions, the designs are generally not feasible for real-world applications. This is due to the assumptions made on the designs (e.g. 2D, having simple geometries), and the trajectories that they are optimized for were derived from simple control policies (e.g. walking on smooth and flat terrain). Purely continuous optimization methods have also been proposed for trajectory and leg design optimization. In [17], such an approach was used to alter design and motion parameters simultaneously using a sensitivity analysis between the parameter sets. Its advantages include faster convergence, better guarantees on optimality and repeatability, and providing deep insight into the design. However, it can only optimize continuous design parameters, which poses a problem should the designer face a discrete set of available hardware.

This work has been conducted as part of ANYmal Research, a community to advance legged robotics. This work was supported in part by the Swiss National Science Foundation through the National Centres of Competence in Research Robotics (NCCR Robotics) and the European Space Agency (ESA), together with Airbus DS in the framework of the Network Partnering Initiative 481-2016.

All the authors are with the Robotic Systems Lab (RSL) at ETH Zurich, Switzerland.

Correspondence should be addressed to Hendrik Kolvenbach (hendrikk@ethz.ch)

Alternatively, the optimization may be formulated as a shortest path problem, where the solution is the simplest combination of modular library components that permits motion tracking [18]. Although it is both efficient and robust, opposite to the continuous approach, it can only optimize discrete design parameters. While the above works are promising, their adoption and application by the research community are ultimately hindered by their closed-source nature.

In this paper, we present the open-source *Matlab* toolbox Vitruvio*. Named after the ancient Roman architect who studied perfect proportion, Vitruvio streamlines the design process by empowering the designer to: i) gain an insight on the robot's performance, ii) compare the effects of different design choices, iii) optimize selected leg design parameters, and iv) visualize the robot. Our framework is versatile from the system to the leg levels. At the former, it can accommodate and analyse a variety of motions and robot designs, such as varying gait patterns, challenging 3D terrains, and a different number of legs. At the latter, the user has full control over the leg's morphology. Example design features include number of links, actuation method (direct or remote using particular actuators), and the integration of elasticities. Unlike previous works, our framework strives to be a realistic and practical tool for the design of a wide variety of robots and motions. Given that the toolbox is open-source, we kept the trajectory input interface generic so that the designer is free to use either a trajectory generation and optimization framework or experimental data from a physical robot. Concerning the optimization, we implemented a simpler derivative-free approach as opposed to a continuous one because our framework is intended for design studies not significantly constrained by computation time, and to allow for the easy integration of additional models. Our framework has been experimentally validated and is relatively accurate (average error per gait cycle of ≈8% in cumulative torque for a 0.36m/s trot for *ANYmal*). With these, Vitruvio presents itself as a highly viable and flexible tool for the design of walking robots, be it for the improvement of existing leg designs with constrained design spaces, or for the free exploration of leg designs for the next generation of walking robots.

II. METHOD

A. Overview

We employ a three-step approach to find optimal leg designs. First, we generate a trajectory for a baseline robot, followed by an analysis of the motion, before finally optimizing the leg with the help of a genetic algorithm (Figure 2).

As we aim for high versatility, our toolbox is independent of the trajectory generation framework to allow for the analysis of motion plans from alternative sources. This separation of the trajectory and design optimization stages requires that the parameters which define the trajectory model are not actively updated in the design optimization loop. As such, the allowable change in mass and inertia of the legs is limited,

*Made available at: https://github.com/leggedrobotics/vitruvio

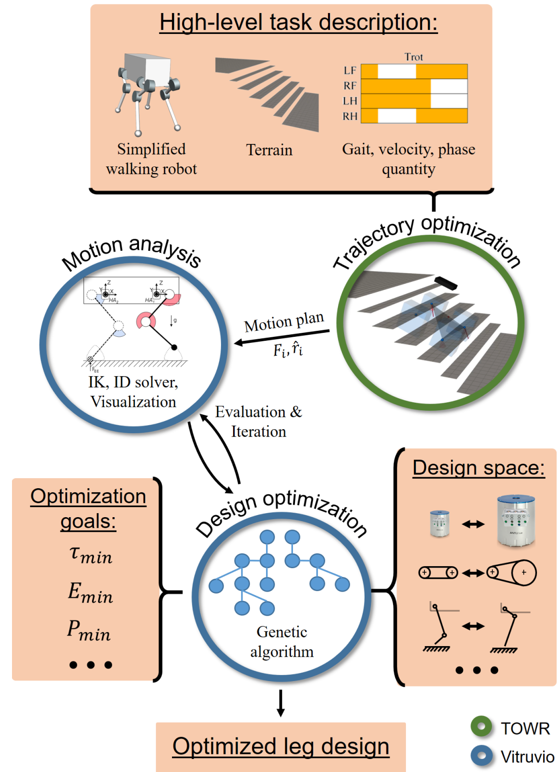

Fig. 2: Overview of the trajectory generation and design optimization method.

and we are not able to observe parameter changes such as robot height, width, length, and range of motion. However, our method allows us to quickly model and analyse a broad range of design parameters and tasks for consideration of vastly different potential designs early in the development phase, while also guiding the more detailed design aspects of the leg.

B. Motion generation

We utilise the single rigid body dynamics (SRBD) trajectory generation framework *TOWR* to generate motion plans for the robots described in this paper. The framework allows us to abstract the task for the user by simply using computer-aided design (CAD) model of the robot, a desired (complex) terrain and preferred gait parameters. A SRBD model is a dynamic model used in trajectory optimization, which is based on centroidal dynamics. Here, the individual rigid bodies of the robot are lumped together into a SRBD model with constant inertia anchored at the center of mass (COM), which is controlled by the contact forces at the end-effectors (EE) [19]. It has been demonstrated in [20] that trajectories generated with this method for the *ANYmal* quadruped could be tracked in simulation as well as directly on the robot. *TOWR* is very versatile as it allows for the generation of highly dynamic motions for a diverse set of legged robots (i.e. monopod, biped, quadruped) on complex terrain.

As inputs to *TOWR*, we provide the robot kinematic model defined by the nominal COM and EE positions and their range of motion; the dynamic model defined by the lumped mass and

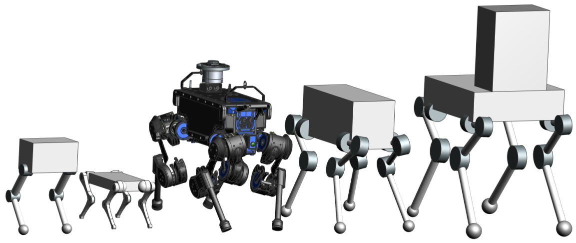

Fig. 3: A set of example robots that can be analysed in Vitruvio, shown alongside *ANYmal* for scale.

centroidal inertia tensor; and the task parameters consisting of the terrain, goal position, duration, gait pattern and the number of phases per leg. Executing the optimization, we obtain the COM and EE trajectories as well as EE forces that satisfy the kinematic and dynamic constraints. However, the following assumptions as stated in [19] have to be met in order to obtain feasible solutions: i) the links are considered as rigid bodies, ii) the momentums of the limbs produced by joint velocities are negligible, and iii) the robot's full body inertia experiences only minor deviations from its nominal value throughout the motion. These assumptions are justified for highly dynamic motions for robots whose limb mass is negligible relative to that of their body. For robots with non-negligible limb mass, the assumptions hold when the movement is very slow or when the limbs experience only a small deviation from their nominal positions [19]. Many existing robots are encompassed by these assumptions including *ANYmal* [5], MIT Cheetah [2], Cassie [21] and HuboDog [22].

We extend this set of justifiable cases to also include small changes to the leg design, so long as these changes result in small deviations from the robot's nominal mass and inertia. This enables us to modify leg design parameters such as link lengths within a window about the nominal design. Design parameters with a small effect on the robot's mass and inertia, such as transmission ratios and spring parameters, can be freely altered without violating any assumptions of the model.

C. Motion analysis

Once the motion has been generated, we analyse the initial guess of the leg which we call the nominal design.

1) Trajectory refinement: First, we apply a series of operations to refine the EE trajectory. To do so, we extract the position of each EE relative to the leg's Hip Abduction/Adduction (HAA) joint. For highly cyclical motions such as trotting on flat ground, we can average multiple steps to create a closed-loop trajectory about the HAA, shown in Figure 4, which reduces computation time. For highly dynamic motions, we interpolate additional points to reduce the gaps between adjacent points due to high EE velocity.

2) Defining leg kinematics: We include the number of links, link lengths, and leg configuration of the initial design to completely describe the model. Using numerical inverse kinematics (IK), we solve for the joint angles required to track the generated EE trajectories with sub-millimeter precision. For a two-link leg, it is sufficient to specify only the link

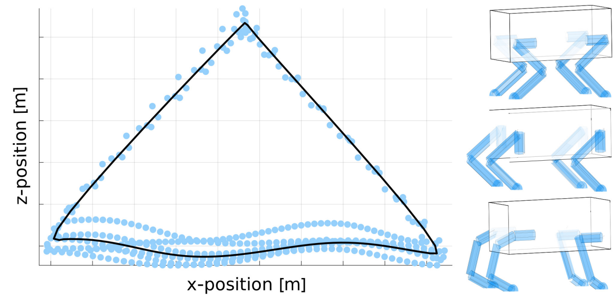

Fig. 4: The motion of an EE relative to its hip attachment. Individual steps (points) are averaged to obtain the EE trajectory over a single step cycle (solid line). Top to bottom: a robot platform with two links, mammal, X-configuration; two links, spider, M-configuration; three links, mammal, M-configuration.

lengths and desired leg configuration (X or M) in order to have a fully defined system. For link quantities greater than two, we can choose to constrain the IK further using heuristics which help to maintain feasibility or allow the algorithm to implicitly minimize joint velocities. Some of the candidate design configurations are illustrated in Figure 4.

3) Joint torque calculation: The final step in the motion analysis is to compute the joint torques necessary to track the motion. We solve for joint torque $\boldsymbol{\tau}$ in stance and swing phase independently,and the case separation is performed based on the ground contact force vector $\mathbf{F}_c$ as seen in Eq.(1). The stance and swing phase models are shown in Figure 5.

$$\boldsymbol{\tau} = \begin{cases} \boldsymbol{\tau}_{stance} & \text{for } \|\mathbf{F}_c\| > 0 \\ \boldsymbol{\tau}_{swing} & \text{for } \|\mathbf{F}_c\| = 0 \end{cases} \tag{1}$$

The simplification has the benefit of allowing us to isolate individual legs for analysis and to change the leg mass without the need to solve the full floating base equations of motion, which would introduce a controller/tracking problem.

The **stance phase** torque is computed with the assumption that dynamic effects are negligible ($\dot{\mathbf{q}}, \ddot{\mathbf{q}} = \mathbf{0}$). This is a reasonable assumption when the motion of the leg is sufficiently slow such that the torque associated with the dynamics is minor relative to the torque due to EE forces. For a trotting motion on the *ANYmal* robot, we find this dynamic torque contribution in stance to be an order of magnitude smaller than the static contribution. For robots with fast-moving, heavy legs, this assumption begins to break down. However, these types of robots are also poorly captured by the set of SRBD assumptions and are thus outside of the scope of this work.

Using a kinetostatic analysis, the stance phase torque is the sum of the joint torque due to the ground reaction forces calculated in *TOWR* and the gravitational forces due to the weight of the leg components. The torque is computed by projecting the contact forces from *TOWR* and the gravitational forces from each of the n components onto the space of generalized coordinates using their respective Jacobians (Eq.(2)).

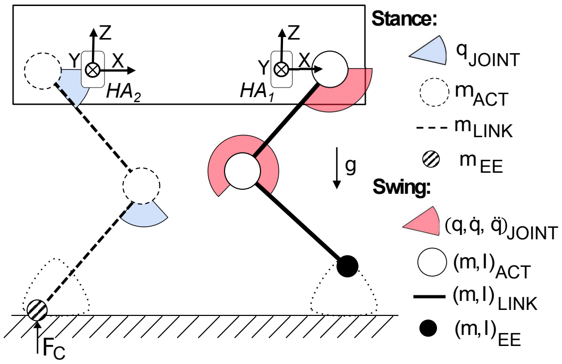

Fig. 5: Stance and swing phase torque contribution model.

Ground reaction forces, $\mathbf{F}_c$ are applied at each EE, and gravitational forces $\mathbf{F}_{grav}$ are applied at the COM of each actuator, link, and EE.

$$\boldsymbol{\tau}_{stance} = \mathbf{J}_c^T \mathbf{F}_c - \sum_{j=1}^{n} \mathbf{J}_j^T \mathbf{F}_{grav,j} \quad (2)$$

During **swing phase**, the relatively high velocity and acceleration motion requires that the momentum of the leg components can no longer be neglected. To accurately compute the torque for tracking the desired EE position in the swing phase, a rigid body tree is created, which connects the hip attachment frame (HA) to the EE by a series of fixed and revolute joints. We apply a linear mass density along each link as well as along the transmission if the joints are remotely actuated. We justify this assumption by the fact that leg links are designed to be lightweight, and the mass of commonly used materials, such as tubes and profiles, scales linearly with length. Point masses are placed at each actuator location and at the EE position to model the weight of the actuators and foot. The joint torques necessary to track the motion are computed using the ID solver from the *Matlab Robotics System Toolbox*† which solves for joint torques using the fixed base dynamics formulation:

$$\boldsymbol{\tau}_{swing} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (3)$$

Where $\mathbf{M}(\mathbf{q})$ is the leg's mass matrix, $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ covering the nonlinear terms (Coriolis, centrifugal) and $\mathbf{g}(\mathbf{q})$ the gravitational terms of the leg.

4) Accounting for changing link mass: During the design optimization stage, described in Sec. II-D, link lengths are continuously changed for each individual i, and the mass of the legs can deviate from the nominal design which was previously used to generate the trajectories. To account for this change in mass Δm during the stance phase, we scale the nominal EE force $\mathbf{F}_{c,nom}$ in relation to the total robot mass M:

$$\mathbf{F}_{c,i} = \left(1 + \frac{\Delta m}{M}\right) \cdot \mathbf{F}_{c,nom} \quad (4)$$

During the swing phase, the change in leg mass and inertia are simply reflected in an updated leg's mass matrix.

†https://ch.mathworks.com/products/robotics.html

5) Model validation: A comparison of the joint torques calculated using this model and using the full ID was conducted for the *ANYmal* robot trotting at 0.1 m/s. The error in cumulative joint torque over the twelve actuators is 2.3% for this motion and there is no significant deviation in the torque shapes for any of the actuators.

Given the instantaneous velocity and torque, we can readily compute the mechanical power as their product and integrate to solve for the mechanical energy.

6) Elasticity, transmission, and motor efficiency models: Having computed the joint level requirements to track the trajectory, we can now optionally apply additional models to propagate the results from joint to actuator and motor levels.

First, we introduce a **parallel elasticities model** to represent torsional springs in parallel with the actuators. The model is based on Hooke's Law and we define each spring such that its deformation angle is equal to that of the joint while assuming this never exceeds the spring's elastic limit. The total joint torque required in tracking the trajectory is the sum of an active torque term τ_{active} provided by the actuator, and a passive torque term τ_{spring} due to spring deformation. This model introduces two additional optimization parameters: the spring constant k and the spring setpoint q_0 (i.e. the joint position when the spring is at its equilibrium position).

$$\tau_{joint} = \tau_{active} + \tau_{spring} \quad \text{and} \quad \tau_{spring} = -k \cdot (q_{joint} - q_0) \quad (5)$$

Next, we introduce a **transmission** model which allows the optimization of the transmission ratio γ. The transmission corresponds to an integrated belt or ball screw drive, between the joint and actuator. The transmission is generally applied at the Knee Flexion/Extension (KFE) joint where it has the additional benefit of reducing the leg inertia.

$$\tau_{actuator} = \frac{\tau_{active}}{\gamma} \quad \text{and} \quad \dot{q}_{actuator} = \gamma \cdot \dot{q}_{joint} \quad (6)$$

This decouples the 1:1 relationship between the actuator and joint velocity and torque and can be included in the optimization to provide an additional degree of freedom, which aids in meeting the joint velocity and torque requirements without exceeding the actuator limits.

Finally, we included a **motor efficiency model** which relates mechanical and electrical power, allowing us to estimate the electrical power and energy consumption of the design. These are factors of particular interest due to their direct influence on the robot's autonomous operating time. Following the method demonstrated in [23], we included a model for generating lookup tables based on motor torque and speed. From those tables, we can obtain the instantaneous operating efficiency and electrical power input to the motors. While integrated, the model has not been validated and is excluded from the current analysis. However, as the toolbox is open-sourced, we hope to extend the database of validated motor efficiency data with the help of the community. Generally, the framework is designed in a modular fashion, where additional modules can be added and toggled on or off depending on the user needs.

D. Leg design optimization

As the formulated problem is both non-linear and non-smooth (because of the possibility to integrate lookup tables), we implement an evolutionary strategy that does not require gradient information to optimize the design parameters. Using this method, an initial set of individuals each with candidate design parameters is created, and their performance is evaluated through a cost function which we seek to minimize. The design parameters of the individuals are modified by the operations of selection, crossover, and mutation over many generations to obtain the design parameters which minimize cost [24]. We isolate a single leg, create the first generation of individuals to uniformly span the design space, and solve for the joint requirements for the current leg design. Each individual is evaluated using a cost function consisting of soft constraints, which help to ensure the physical feasibility of the design, and the user-defined optimization goal. As such, we consider a variety of optimization metrics, such as the minimization of cumulative joint torque, energy, and mechanical cost of transport (MCOT). Note that the total cost of transport (COT) is a measure derived from the average electrical power P_{elec} required to move a robot at constant velocity v at earth gravity g and is widely used for comparing efficiency among robots [25]. Here we consider the mechanical power contribution P_{elec} since the motor efficiency model has not yet been validated:

$$MCOT = \frac{P_{mech}}{mgv} \tag{7}$$

Because we optimize each leg independently, we can only approximate the overall MCOT in the cost function for an individual leg. We estimate this cost using a mass term m that approximates the mass supported by the leg as a function of the leg's mass m_{leg}, trunk mass m_{trunk}, and the number of legs n:

$$m = m_{leg} + \frac{m_{trunk}}{n} \tag{8}$$

We select these as optimization metrics due to the high utility of improvements in these areas (e.g. additional payload capacity, longer autonomous operating time). During the optimization, we do not enforce any symmetry requirements, but due to the nature of the robots and motions that we consider, the optimizer generally finds designs with near left-right symmetry. We allow the optimizer to modify the selected legs within a user-defined range of values for each design parameter. Table I contains the set of candidate user design inputs, optimization parameters and penalty terms that are available to the user. We implemented the genetic algorithm using the *Global Optimization Toolbox*[§] in *Matlab* and we speed up the computation using the *Parallel Computing Toolbox*[¶]. The most time-consuming step is solving the IK, which we speed up by preliminary screening that the leg is long enough to reach the furthest point of the trajectory. Otherwise, the individual is heavily penalized and the remaining steps for the individual are skipped. The optimization is executed on four cores of Intel Core i7 2.5 GHz CPU with 16 GB RAM, and the evaluation of each individual requires on average 0.3 s. The total computation time is dependent on the hyper-parameters of population size and the number of generations. The largest optimizations have a computation time of 2.5 h, but we often get significant cost reductions for smaller rollouts, which require just 15 minutes. Due to the stochastic nature of evolutionary algorithms, multiple optimization trials subject to the same conditions do not always yield the same result. We find that consistent results are obtained for population size and the number of generations greater than 150 and 100, respectively. It is important to note that the results of the optimization are highly task-specific, meaning that the design is optimal for a specific motion and may be sub-optimal or even infeasible for other motions. We can guarantee versatility by providing a longer trajectory which includes common obstacles such as steps and test the ability of the task-specific design to navigate this varied task. Alternatively, we can directly optimize the design for the varied task to obtain a more well-rounded design.

Robot design inputs	
Property	**Options**
Leg layout	Spider, mammal
Leg configuration	X, M
Number of links	[2,4]
Actuator selection	ANYdrive, Dynamixel, Neo[‡]
Actuation method	Direct, remote
Parallel elasticities	Included, excluded
Penalty terms	
Soft constraints	**Optimization metrics**
Limit allowable leg extension	Min. joint/actuator velocity [rad/s]
Enforce actuator limits	Min. joint/actuator torque [Nm]
Joints are above the ground	Min. mech./elec. power [W]
Penalize tracking error	Min. mech./elec. energy [J]
Prevent leg collisions	Min. antagonistic power [W]
	Min. mechanical COT
	Max. motor operating efficiency
Optimization Parameters	
Link lengths [m]	
Transmission ratio [-]	
Spring constant [Nm/rad]	
Spring set point [rad]	

TABLE I: Candidate inputs, constraints, penalties and parameters for the design optimization.

III. RESULTS AND DISCUSSION

A. Validation by comparison of simulated and measured data

We now seek to validate the Vitruvio pipeline from simplified CAD model through to the joint torque and energy calculation for the nominal design. For this comparison, we consider constant trotting at 0.36 m/s on flat ground and compare the simulated data with measured data captured on *ANYmal*. By validating the feasibility of the pipeline for this nominal case, we reason that similar topologies and motions can be accurately simulated.

[§]https://www.mathworks.com/products/global-optimization.html
[¶]https://ch.mathworks.com/products/parallel-computing.html

Joint	Average RMSE		Average Error Per Gait Cycle	
	Velocity [rad/s]	Torque [Nm]	Cumulative Torque	Cumulative Mech. Energy
Hip Abduction/Adduction (HAA)	0.66	6.84	-7.85%	+1.07%
Hip Flexion/Extension (HFE)	1.45	10.86		
Knee Flexion/Extension (KFE)	2.23	9.17		

TABLE II: Comparison of the simulated and measured results.

To generate the kinematic and dynamic model of the robot, we use the CAD model shown to the right of *ANYmal* in Figure 3. The measured data is plotted against the simulated data in Figure 6. Due to near left-right symmetry, only the data for the right front (RF) and right hind (RH) legs are displayed.

We compare the results of our simulation with the measured data based on joint torque, velocity, cumulative joint torque, and mechanical energy consumption of all twelve actuators. The accuracy of the simulation is summarised in Table II and shows a relatively low error of approximately 8% for cumulative joint torque and 1% for the cumulative mechanical energy.

B. Leg design optimization for an existing robot

Having verified the simulation framework for the nominal design, we apply the optimization to redesign the *ANYmal* legs. The lengths of the thighs and shanks are optimized within a window about the nominal design defined by:

$$0.5 \cdot \mathbf{L}_{link,nom} \leq \mathbf{L}_{link,opt} \leq 2 \cdot \mathbf{L}_{link,nom} \qquad (9)$$

The high-level design aspects are fixed for the *ANYmal* platform, meaning that each leg consists of two links directly actuated using three *ANYdrive* actuators. The leg layout and configuration are set to mammal and X-configuration respectively. Soft constraints listed in Table I ensure the physical feasibility of the design, but hardware limits are not imposed. We provide the same motion task as presented in III-A and the optimization is run for each, the minimization of cumulative joint torque, mechanical energy, and mechanical cost of transport. The results of the design optimization for mechanical energy minimization are plotted alongside the measured data and nominal design simulation in Figure 6, and the key results are compiled in Table III.

For the cumulative joint torque minimization case, the optimized design experiences a significant reduction in thigh lengths and increase in shank lengths causing a a decrease in total robot mass of 1.7 kg or 4.4%. This trend, although less significant, is also observed by [14]. The increased shank length results in a torque reduction by decreasing the moment arm at the KFE joint. While this change reduces peak torque, it increases peak velocity. The overall result is a reduction in cumulative joint torque of 10.3%. The optimal designs for minimization of mechanical energy and MCOT are very similar with a significant increase in thigh lengths and slight decrease in shank lengths.

As a result, the total robot mass increases by 2.0-2.5 kg or 5-6% for these two optimized designs. This overall increase in limb length results in two opposing phenomena. First, the joint velocities decrease, which reduces the power requirement

Metric of Comparison	Component			
	Thigh		Shank	
Link Lengths [mm]	**Front**	**Hind**	**Front**	**Hind**
Nominal design	250	250	330	330
Joint torque minimization	126	129	456	421
Mech. energy minimization	384	376	256	267
MCOT minimization	412	404	261	279
Robot Mass	**Entire Robot**			
Nominal design	38.8 kg			
Joint torque minimization	37.1 kg			
Mech. energy minimization	40.8 kg			
MCOT minimization	41.3 kg			
Improvement in Optimization Metric				
Joint torque minimization	10.3%			
Mech. energy minimization	5.0%			
MCOT minimization	9.3%			

TABLE III: Results of the *ANYmal* link length optimization.

and, therefore, the energy consumption of the legs. At the same time, the joint torques increase, which acts to increase the power and energy requirements. Here, the effect of the joint velocity reduction is dominant and the optimization results in a mechanical energy reduction of 5.0% and MCOT reduction of 9.3% relative to the nominal design.

These examples show that a significant reduction in the optimization goal metric is possible by optimization of the link lengths, but this may have detrimental effects on other performance metrics. In a general sense, choosing equal link lengths offers a good balance for a "universal" type platform such as *ANYmal*.

C. Leg design optimization for new robot platforms

In addition to *ANYmal*, we also apply our framework to the analysis and optimization of a set of virtual robots. Trajectories are generated for tasks which include trotting, hopping, pronking, and stair climbing. We create the nominal design with equal link lengths and a transmission ratio of one. When applicable, we make a first guess at the optimal spring constant and we set the spring engagement point to the mean joint position. The framework is then used to optimize the set of design parameters to minimize cumulative mechanical actuator energy and cumulative actuator torque. For the stair climbing motion, the results for the left hind (LH) legs are also displayed due to significant front-hind asymmetry. Otherwise, only the LF leg is considered. The results are listed in Table IV while the robots are visualized in Figure 8.

The key findings are the following. First, energy-minimized designs prefer longer links with higher stiffness springs, whereas actuator torque-minimized designs prefer short links and lower spring stiffness. This is consistent with the trend observed in the *ANYmal* optimization in Section III-B. Second, the inclusion of the spring at the KFE joint allows for much greater reduction in actuator torque and energy than is possible with just link length optimization. Finally, the transmission ratio parameter is effective in maintaining the operating region within the actuator limits. This result is observed for the actuator torque-optimized designs with $\gamma < 1$.

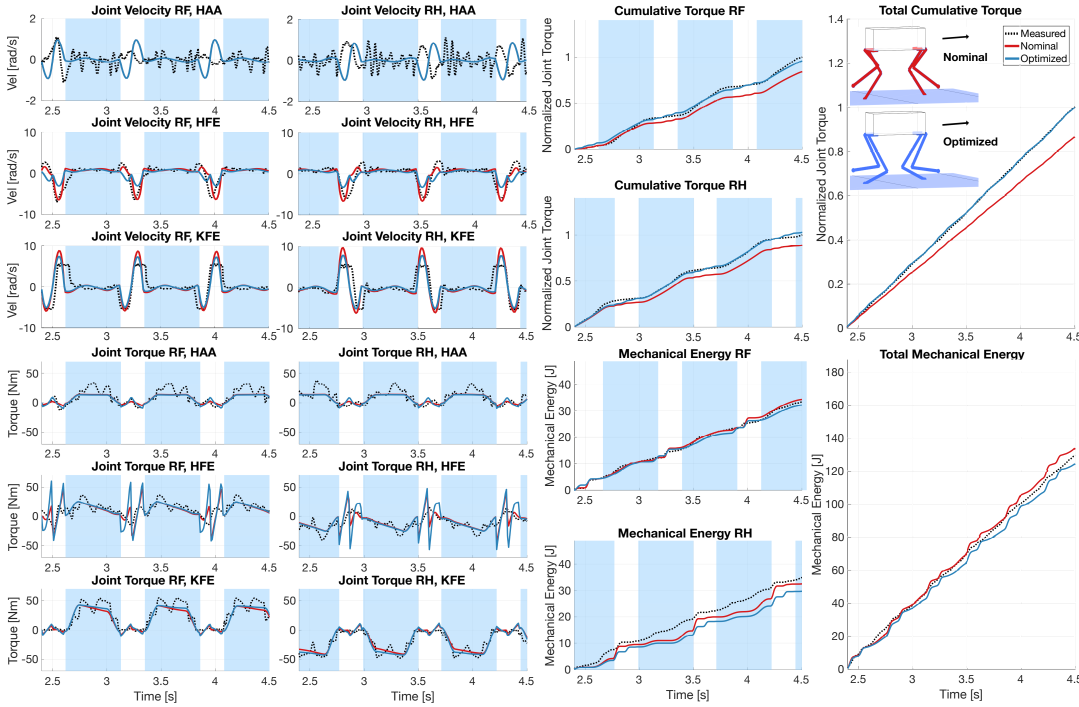

Fig. 6: Comparison of the measured *ANYmal* data against the simulated results for the existing and energy-optimized designs.

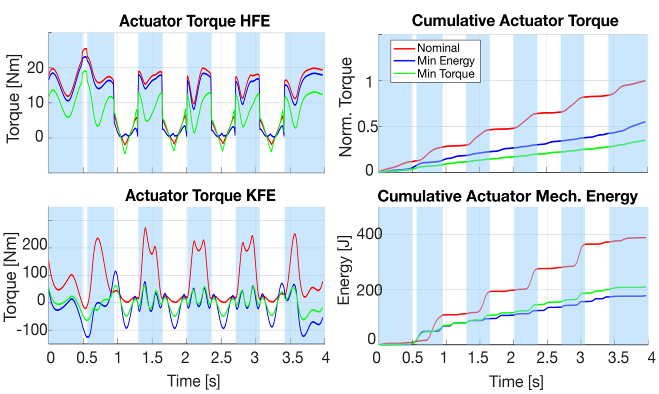

Fig. 7: Comparison of the nominal, actuator energy minimized and actuator torque minimized designs of a 40kg monopod robot.

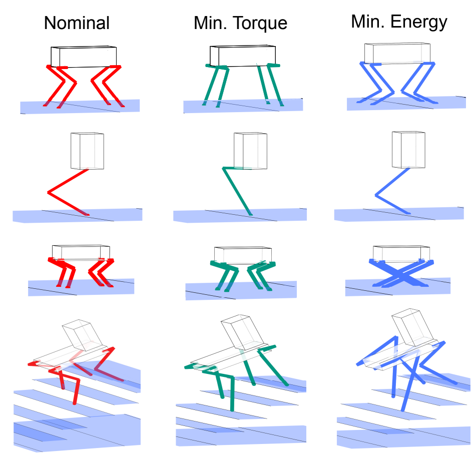

Fig. 8: Nominal and optimized designs for *ANYmal* and hypothetical platforms. Top to bottom: *ANYmal*, trot; 40 kg monopod, hop; 10 kg quadruped, pronk; 80 kg quadruped, stair climb.

Intuitively, the transmission ratio should be greater than one to reduce the actuator torque for these cases, however, the transmission ratio instead acts to increase the actuator torque in order to reduce the actuator velocity below the hardware limits. The results for optimized energy consumption and torque reduction of the hopping monopod robot are plotted in Figure 7.

IV. CONCLUSION

We have developed and open-sourced a robot leg design toolbox for *Matlab* that guides designers from the initial conception of a robot's geometry and application, through the analysis of high-level design choices, to the optimization of leg designs. This toolbox is applicable to diverse robot topologies as long as their motions fall within the assumptions of the single rigid body dyanmics model. We can optimize the leg design based on metrics such as torque and energy minimization while ensuring the feasibility of the generated design, subject to hardware limitations imposed by the actuators. Validation with measured data from *ANYmal* shows that actuator requirements are forecasted relatively accurately with respective errors in mechanical energy consumption and cumulative joint torque of 1% and 8% per gait cycle for a typical trot. Optimization of the *ANYmal* link lengths for this motion indicates that it is possible to reduce cumulative joint torque, mechanical energy, and MCOT by approximately 5-10%. When considering the design of novel robots, the toolbox aids in generating feasible and optimized leg designs taking into account link lengths, transmission ratios, and spring parameters as design parameters.

	Link Length [mm]		Transmission KFE	Spring KFE		Optimization Metric	
	Thigh	Shank	γ	k [Nm/rad]	q0 [rad]	$\triangle$ Cum. Actuator Torque [%]	$\triangle$ Cum. Actuator Energy [%]
40kg monopod, hop							
Nom	500	500	1.00	100	-1.89	-	-
Min. Energy	561	414	0.99	369	-1.65	-46	-54
Min. Torque	301	621	0.75	199	-1.68	-65	-46
10kg quadruped, pronk							
Nom LF	150	150	1.00	20	-1.59	-	-
Min. Energy LF	260	207	1.00	46	-2.16	+45	-19
Min. Torque LF	153	168	0.81	17	-1.61	-9	-14
80kg quadruped, stairs							
Nom LF (LH)	400 (400)	400 (400)	1.00 (1.00)	-	-	-	-
Min. Energy LF (LH)	428 (554)	348 (790)	1.00 (0.92)	-	-	-5 (+14)	-2 (-8)
Min. Torque LF (LH)	200 (336)	575 (455)	2.00 (2.00)	-	-	-32 (-10)	+18 (+6)

TABLE IV: Optimization results for hypothetical robot platforms.

While the leg optimization toolbox proved useful in the development of optimized legged robots, it is limited in the following ways: i) the optimized designs are tailored to the given motion task which requires re-running the tool multiple times should a number of tasks need to be fulfilled, ii) the allowable change of the leg mass and inertia is limited to prevent significant deviation from the values used in the trajectory generation phase, iii) the actuator requirements are sensitive to the nominal stance provided as input for the trajectory generation, which is influenced by the design but is also closely tied to the control of the robot, and iv) when searching for the optimal design for a single metric such as joint torque or mechanical energy, the optimal design may be found to be on one extreme or the other and a well-rounded, practical design likely exists somewhere in between. Overcoming this requires tuning multiple weight terms against one another in the cost function.

Future work should seek to extend the toolbox. The motor efficiency model needs to be validated to enable the energetic COT optimization. Additional discrete design parameters, such as leg configuration and number of links, can also be implemented as optimization parameters.

ACKNOWLEDGEMENT

We would like to thank Alexander Winkler for his guidance in applying *TOWR* and Ruben Grandia for his support in the motion analysis section of this work.

REFERENCES

[1] A. Ananthanarayanan, M. Azadi, and S. Kim, "Towards a bio-inspired leg design for high-speed running," *Bioinspiration & Biomimetics*, vol. 7, p. 046005, 08 2012.

[2] S. Seok, *et al.*, "Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117–1129, June 2015.

[3] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, 07 2013.

[4] 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 *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 483–490.

[5] M. Hutter, *et al.*, "ANYmal - toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[6] 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*, vol. 225, no. 6, pp. 831–849, 2011.

[7] S. Hirose, *et al.*, "Quadruped walking robots at tokyo institute of technology," pp. 104–114, June 2009.

[8] A. Abate, J. Hurst, and R. L. Hatton, "Mechanical antagonism in legged robots," 06 2016.

[9] J.-W. Chung, I.-W. Park, and J.-H. Oh, "On the design and development of a quadruped robot platform," *Advanced Robotics*, vol. 24, no. 1-2, pp. 277–298, 2010.

[10] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design: Modeling and Computation*, 3rd ed. Cambridge University Press, 2017.

[11] K. Sims, "Evolving virtual creatures," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994, pp. 15–22.

[12] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[13] K. M. Digumarti, C. Gehring, S. Coros, J. Hwangbo, and R. Siegwart, "Concurrent optimization of mechanical design and locomotion control of a legged robot," in *Mobile Service Robotics*. World Scientific, 2014, pp. 315–323.

[14] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Task-based limb optimization for legged robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2062–2068.

[15] M. F. Silva, "Quadruped robot optimization using a genetic algorithm," in *Field Robotics*. World Scientific, 2012, pp. 782–789.

[16] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 60.

[17] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[18] S. Ha, S. Coros, A. Alspach, J. M. Bern, J. Kim, and K. Yamane, "Computational design of robotic devices from high-level motion specifications," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1240–1251, 2018.

[19] A. W. Winkler, "Optimization-based motion planning for legged robots," Ph.D. dissertation, ETH Zurich, 2018.

[20] A. W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, pp. 1560–1567, July 2018.

[21] A. M. Abate, "Mechanical design for robot locomotion," 2018.

[22] J.-W. Chung, I.-W. Park, and J.-H. Oh, "On the design and development of a quadruped robot platform," *Advanced Robotics*, vol. 24, no. 1-2, pp. 277–298, 2010.

[23] A. Mahmoudi, W. L. Soong, G. Pellegrino, and E. Armando, "Efficiency maps of electrical machines," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2015, pp. 2791–2799.

[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[25] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.