Control-Aware Design Optimization for Bio-Inspired Quadruped Robots

Flavio De Vincenti, Dongho Kang, Stelian Coros

Abstract— **We present a control-aware design optimization method for quadrupedal robots. In particular, we show that it is possible to analytically differentiate typical, inverse dynamics-based whole body controllers with respect to design parameters, and that gradient-based methods can be used to efficiently improve an initial morphological design according to well-established metrics.**

We apply our design optimization method to various types of quadrupedal robots, including designs that feature closed kinematic chains. The methodology we present enables a principled comparison of different types of optimized legged robot designs. Our experiments, for example, suggest that mechanically-coupled three-link leg designs present notable advantages in terms of performance and efficiency over the common two-link leg designs used in most quadrupedal robots today.

I. INTRODUCTION

The animal kingdom has always inspired the design of legged robots and the algorithms to control them [1, 2, 3, 4]. This fact is rooted in the numerous challenges inherent to these machines. Indeed, locomotion is a complicated task, requiring the simultaneous coordination of different apparatuses. All body parts must be functional and have to collectively support the movements of the whole system.

Once a machine is built, control specialists take the hardware to the limit and assess its performance. The collected data inform the construction of the next prototype, and the whole process is repeated in an iterative fashion. This approach to mechanical design not only demands considerable resources, but it frequently ends up with sub-optimal results [5]. For this reason, design optimization methods have become increasingly important.

In this paper, we introduce computational models aimed at significantly reducing these manual efforts and improving their outcomes. We use our optimization pipeline to enhance the hind leg architecture of a bio-inspired quadruped robot (see Fig. 1). In widely adopted approaches [5, 6, 7, 8], a design is evaluated based on open-loop trajectories output by a kinodynamic planner. In contrast, we augment the optimization loop with a differentiable simulator and a whole-body controller (see Fig. 2). Although this device increases the computational burden, it allows a design to evolve in a more realistic setting and instills awareness of the control laws in it. To assess a model, we compute a performance measure over a given set of control tasks in simulation. Finally, we feed the gradient of this index to a quasi-Newton method that updates the design and initiates a new iteration.

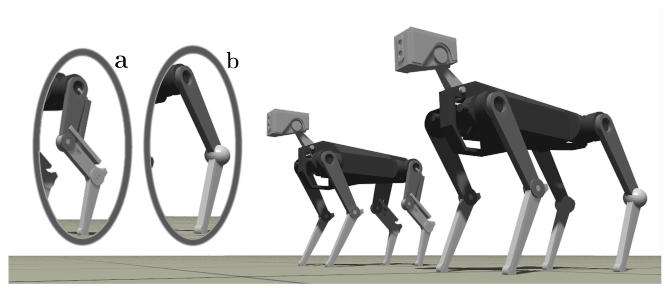

Fig. 1. The robots *Dog V3* (left) and *Dog V2* (right). The design of *Dog V3*'s hind limbs is inspired by the anatomy of quadruped mammalians (**a**), whereas *Dog V2* carries standard two-link legs (**b**). In this work, we optimize their designs and compare their performances over a diverse set of motion tasks.

We adapt link lengths and closed kinematic chain (CKC) morphologies in both two- and three-segmented leg designs. To this end, we formulate a holonomic constraint to model general planar closed-loop topologies, and we extend our whole-body control (WBC) implementation accordingly.

In summary, we present the following contributions:

- We propose a design optimization methodology that takes into consideration the control objectives of a WBC implementation. Our pipeline can adapt any continuous variables a robotic design may depend on using a gradient method. We further derive the analytical expressions of the required derivatives via SA.
- To demonstrate our approach, we optimize the three-segmented hind limbs of a bio-inspired, four-legged robot constrained by planar CKCs. Specifically, we optimize for link lengths and relative orientations between rigidly attached bodies.
- We verify in simulation that optimized three-segmented leg designs have clear advantages over standard two-link ones with respect to various well-established performance metrics.

II. RELATED WORK

A. Design Optimization Methods

Design optimization problems are concerned with finding the best parameters that fully specify a mechanical system based on some performance metrics. The design space may consist of continuous variables (e.g., the length of a link) or discrete decision variables (e.g., the type of an actuator). The former setting is more common and is typically approached by using stochastic, derivative-free methods, such as covariance matrix adaptation evolution strategy (CMA-ES). Digumarti et al. [9] used this algorithm to determine

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866480).

The authors are with the Computational Robotics Lab, ETH, Zurich, Switzerland. dflavio@ethz.ch

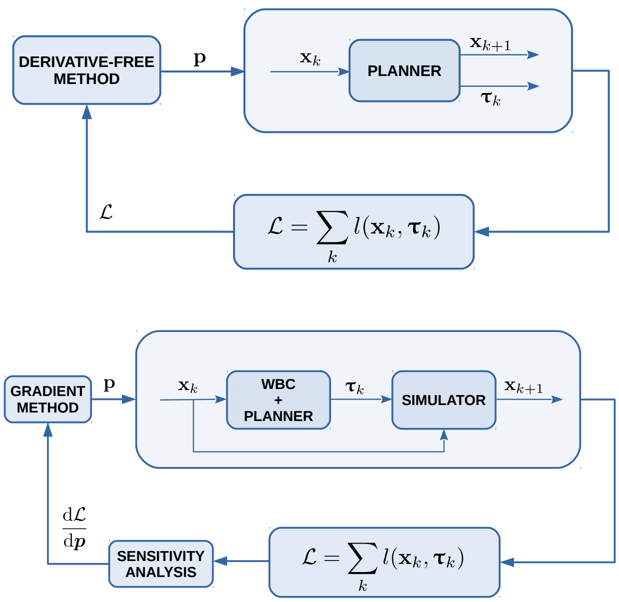

Fig. 2. Comparison of general-purpose design optimization methodologies. **Top:** Most existing approaches adopt stochastic, derivative-free methods to tackle design optimization problems [5, 7, 9]. At each iteration, the optimization algorithm updates the design parameters $\mathbf{p}$ based on a performance measure $\mathcal{L}$ that assesses the open-loop states $\mathbf{x}_k$ and control inputs $\boldsymbol{\tau}_k$ generated by a kinodynamic planner. **Bottom:** Our approach augments the optimization loop with a QP-based whole-body controller and a differentiable simulator. We propose computing the gradient of the objective function efficiently through SA and feeding it to a gradient method.

both the optimal morphology and controller parameters to maximize the speed of a quadruped robot subject to actuator limits. Ha et al. [5] employed CMA-ES to find the best link lengths for different legged machines to perform a single, predetermined task. They noticed that three-segmented legs lessen the torque requirements of a robot because they afford shorter moment arms. However, since they actuated all the joints, they unnecessarily increased their power consumption and inferred the supremacy of two-link designs over three-link ones.

Chadwick et al. [7] proposed a computational framework that optimizes for the legs of walking robots using a genetic algorithm (GA). It takes into account lengths and masses of the links, torsional spring parameters, and efficiency models for the motors. The motion trajectories are computed by solving a nonlinear programming problem (NLP), and each design is evaluated based on the corresponding open-loop joint torques.

Ha et al. [6] adopted SA to simultaneously optimize design and motion parameters with a gradient method. Similarly, Dinev et al. [8] used gradient descent in a bilevel optimization strategy to adapt link lengths and base dimensions for a quadruped robot. They relied on numerical differentiation to obtain the derivatives of a motion planner formulated as an NLP. In this work, we also present an SA formulation and employ finite difference (FD) to compute the required gradients. However, we include a whole-body controller in the optimization loop: this addition encourages more refined results, as it makes the procedure closer to real-world experimental setups. Moreover, we optimize for complex four- and five-bar linkages that constrain the motion of the system in a nontrivial way.

Finally, Chevallereau et al. [10] proposed a method to guide the leg design of bipeds with mono-articular and bi-articular linear actuation. Their approach includes a first, analytic-heuristic step that provides insight into the design problem, and a second, numerical one that uses the sequential quadratic programming (SQP) algorithm. Once again, the performance of a design is evaluated based on open-loop signals as opposed to our methodology.

B. Whole-Body Control Algorithms for Legged Robots

Bellicoso et al. [11] formulated an inverse dynamics controller for *ANYmal* [12] as a cascade of quadratic programming (QP) problems; these were solved hierarchically to enforce strict priorities on the various tasks to be performed. In [13], the authors further enhanced their implementation with a planner based on the zero-moment point (ZMP). For all our experiments, we employed a similar QP-based control strategy, but without strictly prioritizing the various objectives. This decision facilitated our implementation and did not affect the motion capabilities of the systems we considered.

Recently, the WBC framework has also been studied in the context of model predictive control (MPC) formulations where the full dynamics of complex systems are taken into account [14] or, alternatively, simplified models are used in combination with hierarchical whole-body controllers. Kim et al. [15] applied the latter approach to the quadruped robot *Mini-Cheetah* [16]; convex MPC was used to find optimal reaction forces with a simple model of the robot, and whole-body hierarchical optimization was employed to compute joint torque, position and velocity commands based on the reaction forces computed by MPC.

In principle, any control algorithm can be integrated into our methodology, as long as it is differentiable – for instance, [11, 15, 17]. We adopted a QP-based whole-body controller because it represents the best compromise between complexity and performance, and its derivatives can be computed through closed-form expressions [18].

C. Bio-Inspired Mechanical Legs

Two-link leg designs have become the standard choice for quadruped robots. However, the anatomy of cats and dogs exhibits three-link legs comprising a femur, a tibia, and a foot segment [19]. This fact motivated a breed of bio-inspired four-legged machines, such as *BigDog* [2], *MIT Cheetah* [3], and *Cheetah-Cub* [4]. Previous work highlighted the advantages of bio-inspired leg designs. However, to the best of our knowledge, there exists no study comparing the performance of two- and three-segmented legs over generic control tasks.

Ananthanarayanan et al. [20] showed the efficacy of tendon-bone co-location designs in reducing the stresses acting on the limbs of *MIT Cheetah* [3]. They further proposed a novel fabrication method to produce links with structural properties peculiar to mammalian bones. Ruppert and Badri-Spröwitz [19] tested a biarticular muscle-tendon structure in a robotic leg and verified its superiority to conventional pantograph architectures in terms of energy efficiency. Similar experiments and conclusions were attained for various other bio-inspired leg designs – e.g., [21, 22].

All the above designs were conceived after careful inspections of animal anatomies. However, their performance analyses were mostly carried out on simplified experimental setups or relied on stripped-down simulation models. Also, because their focus was rather specific, the authors did not propose original optimization tools to tackle general design problems.

III. WHOLE-BODY CONTROLLER

In this section, we describe a WBC formulation that implements the inverse dynamics of constrained floating-base robots as the solution of a QP problem [11]. This controller is qualified to accommodate dynamic motion tasks such as trotting or jumping. Furthermore, under mild requirements on the definition of the QP control problem, the computed solution can be differentiated with respect to any parameter the system may depend on [18]. In combination with a differentiable simulator and SA [23], a robot becomes amenable to undergoing design optimization via gradient methods.

A. Constrained Mechanical Systems

The equations of motion [24] of a constrained legged robot can be written as:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{u}} + \mathbf{h}(\mathbf{q},\mathbf{u}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q},\mathbf{u})^\top \boldsymbol{\lambda} \tag{1a}$$

$$\text{s.t.} \quad \mathbf{J}_c(\mathbf{q},\mathbf{u})\dot{\mathbf{u}} = \mathbf{w}(\mathbf{q},\mathbf{u})\,, \tag{1b}$$

where $\mathbf{q} \in \mathbb{R}^{n_q}$ is the vector of the generalized coordinates, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of the generalized velocities, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_u \times n_u}$ is the symmetric and positive definite inertia matrix, $\mathbf{h}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{n_u}$ is a generalized force vector containing the Coriolis, centrifugal and gravitational effects, $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$ is the actuation vector, and $\mathbf{S} \in \{0,1\}^{n_\tau \times n_u}$ is the joint selection matrix that defines how the joint torques act on the system. Finally, $\mathbf{J}_c(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{n_c \times n_u}$ is the Jacobian associated with n_c constraints, $\boldsymbol{\lambda} \in \mathbb{R}^{n_c}$ is the vector of the Lagrange multipliers corresponding to the constraint forces, and $\mathbf{w}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{n_c}$ is an arbitrary function of $\mathbf{q}$ and $\mathbf{u}$.

The constraint equation (1b) is written in the so-called acceleration form, and its rows may enforce, among others, contact constraints or kinematic restrictions due to closed-loop topologies. If the support feet of the robot are modeled as points [11, 24], then each contact constraint introduces three equations of the form ${}_{\mathcal{I}}\mathbf{r}_{IS}(\mathbf{q}) = const. \in \mathbb{R}^3$, which can be differentiated twice to yield

$$\mathbf{J}_S(\mathbf{q})\dot{\mathbf{u}} = -\dot{\mathbf{J}}_S(\mathbf{q},\mathbf{u})\mathbf{u}\,. \tag{2}$$

In the next section, we express arbitrary planar CKC constraints in a parameterized, general form.

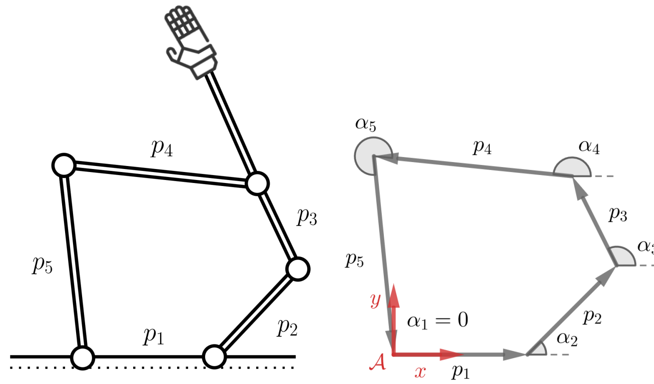

Fig. 3. Example of a robotic arm featuring a five-bar linkage. **Left:** Graphical depiction of a manipulator with $n_p = 5$ consecutive links of lengths $p_1, p_2, \ldots, p_5$. **Right:** Geometric representation of the planar CKC. Each link i is construed as a vector with radial and angular coordinates p_i and α_i, respectively, relative to the reference frame $\mathcal{A}$ for $i = 1, 2, \ldots, 5$.

B. Planar Closed Kinematic Chains

In Section V, we optimize the hind leg design of a quadruped robot featuring both four- and five-bar linkages by using a quasi-Newton method. To do so, we define each CKC as a differentiable scleronomic constraint depending on a set of continuous design parameters.

Apart from merely aesthetic considerations, CKCs in a legged robot are convenient for different reasons. Firstly, they allow conveying torque signals to the passive joints of a robot without resorting to additional actuators [3]. Moreover, by arranging all the motors on the robot body, the legs can be made significantly lightweight, which is a requirement for many quadrupedal system control algorithms [15]. Finally, closed-loop topologies are found in animal legs in the form of muscle-tendon structures, which demonstrated both structural and mechanical benefits [19, 20, 21, 22].

Let $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \cdots & p_{n_p} \end{bmatrix}^\top \in \mathbb{R}^{n_p}$ denote the vector of design parameters that fully describe a planar CKC with n_p consecutive links. Specifically, let $p_1, p_2, \ldots, p_{n_p}$ be the lengths of these links. Then, the closed-loop topology constrains the origin of the first link along the opened mechanism to coincide with the tip of the final one (see Fig. 3).

Under typical definitions of the vector of the generalized coordinates $\mathbf{q}$, we can express a planar CKC constraint as:

$$\mathbf{f}_{\text{CKC}}(\mathbf{q},\mathbf{p}) \coloneqq \begin{bmatrix} \mathbf{cos}\,(\boldsymbol{\Theta}\mathbf{q} + \boldsymbol{\theta})^\top \\ \mathbf{sin}\,(\boldsymbol{\Theta}\mathbf{q} + \boldsymbol{\theta})^\top \end{bmatrix} \mathbf{p} = \mathbf{0}_{2\times 1}\,, \tag{3}$$

where $\boldsymbol{\Theta} \in \{0,1\}^{n_p \times n_q}$ is a constant binary matrix, $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ is a constant vector of angle offsets, and $\mathbf{cos} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{sin} : \mathbb{R}^n \to \mathbb{R}^n$ represent the element-wise cosine and sine functions, respectively; $\mathbf{0}_{m\times n} \in \mathbb{R}^{m \times n}$ is the $m \times n$ zero matrix. In turn, we can rewrite (3) in the acceleration form (1b) by differentiating it twice, resulting in:

$$\mathbf{J}_{\text{CKC}}(\mathbf{q},\mathbf{p})\dot{\mathbf{u}} = -\dot{\mathbf{J}}_{\text{CKC}}(\mathbf{q},\mathbf{u},\mathbf{p})\mathbf{u}\,, \tag{4}$$

where $\mathbf{J}_{\text{CKC}}(\mathbf{q},\mathbf{p}) \in \mathbb{R}^{2\times n_u}$ is the planar CKC constraint Jacobian matrix.

C. Quadratic Programming Control Problem

One of the simplest yet most effective ways to simultaneously control different tasks for a constrained robotic system involves solving a QP problem [11]. Specifically, the QP control problem at a certain time t given the vectors $\mathbf{q}$ and $\mathbf{u}$ of the generalized coordinates and velocities, respectively, has the following form:

$$\arg\min_{\mathbf{z}} \ \frac{1}{2}\mathbf{z}^\top \mathbf{Q}(\mathbf{q},\mathbf{u})\mathbf{z} + \mathbf{c}(\mathbf{q},\mathbf{u})^\top \mathbf{z} \tag{5a}$$

$$\text{s.t. } \mathbf{A}(\mathbf{q},\mathbf{u})\mathbf{z} = \mathbf{b}(\mathbf{q},\mathbf{u})\,, \tag{5b}$$

$$\mathbf{K}(\mathbf{q},\mathbf{u})\mathbf{z} \preceq \mathbf{l}(\mathbf{q},\mathbf{u})\,, \tag{5c}$$

where $\mathbf{z} \in \mathbb{R}^n$, $\mathbf{Q}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{n\times n}$, $\mathbf{c}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^n$, $\mathbf{A}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{m\times n}$, $\mathbf{b}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^m$, $\mathbf{K}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^{p\times n}$, $\mathbf{l}(\mathbf{q},\mathbf{u}) \in \mathbb{R}^p$, and the symbol $\preceq$ denotes a component-wise non-strict inequality. For the above problem to be convex and differentiable, $\mathbf{Q}(\mathbf{q},\mathbf{u})$ must be symmetric and positive definite, and the rows of $\mathbf{A}(\mathbf{q},\mathbf{u})$ must be linearly independent [18]. The vector of the optimization variables $\mathbf{z}$ can be chosen in a number of different ways, each affecting the makeup of the QP control problem, as well as the computational effort required to solve it. In Section V, we conduct our analysis having defined $\mathbf{z} := \begin{bmatrix}\dot{\mathbf{u}}^\top & \boldsymbol{\tau}^\top & \boldsymbol{\lambda}^\top\end{bmatrix}^\top \in \mathbb{R}^{n_u+n_\tau+n_c} = \mathbb{R}^{n_z}$. This choice greatly simplifies the formulation of the controller. Indeed, the plain equations of motion (1) of the constrained system can be directly translated to the form of the equality constraints (5b) as:

$$\begin{bmatrix}\mathbf{M}(\mathbf{q}) & -\mathbf{S}^\top & -\mathbf{J}_c(\mathbf{q},\mathbf{u})^\top \\ \mathbf{J}_c(\mathbf{q},\mathbf{u}) & \mathbf{0}_{n_c\times n_\tau} & \mathbf{0}_{n_c\times n_c}\end{bmatrix}\mathbf{z} = \begin{bmatrix}-\mathbf{h}(\mathbf{q},\mathbf{u}) \\ \mathbf{w}(\mathbf{q},\mathbf{u})\end{bmatrix}\,. \tag{6}$$

In our implementation, we assume that the magnitude of each joint torque must be less than or equal to some constant $\tilde{\tau} \in \mathbb{R}_{>0}$. Thus, we can write a linear inequality of the form (5c) as:

$$\begin{bmatrix}\mathbf{0}_{n_\tau\times n_u} & \mathbf{I}_{n_\tau} & \mathbf{0}_{n_\tau\times n_c} \\ \mathbf{0}_{n_\tau\times n_u} & -\mathbf{I}_{n_\tau} & \mathbf{0}_{n_\tau\times n_c}\end{bmatrix}\mathbf{z} \preceq \tilde{\tau}\,\mathbf{1}_{(2\times n_\tau)\times 1}\,, \tag{7}$$

where $\mathbf{1}_{m\times n} \in \mathbb{R}^{m\times n}$ is the $m\times n$ matrix whose entries are all equal to 1, and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the $n\times n$ identity matrix.

We additionally impose a pyramidal approximation of the friction cone constraints on the ground reaction forces $\boldsymbol{\lambda}_c$ as presented in [11, eq. (29)]. In all our experiments, the robot treads on a flat ground with constant surface normal $\mathbf{n} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^\top \in \mathbb{R}^3$.

Finally, the objective function (5a) prescribes swing leg and base motion tracking tasks, contact foot position tracking tasks, and overall joint torque minimization tasks. All the target trajectories for the various control tasks are computed by a locomotion planner, which we introduce in Section V.

IV. DIRECT SENSITIVITY ANALYSIS

We evaluate a robotic design by calculating a performance measure over a fixed set of control experiments in simulation [5, 6, 7] (see Section V). For the sake of simplicity, let us consider a scenario where the CKC design must be optimized over a single task – the same formulation can then be straightforwardly extended to any number of trials. Moreover, let the task of interest span N discrete time steps of duration $h \in \mathbb{R}_{>0}$.

Let $\mathbf{x}_k := [\mathbf{q}_k^\top\ \mathbf{u}_k^\top]^\top \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$ and $\boldsymbol{\tau}_k \in \mathbb{R}^{n_\tau}$ represent the state vector and the actuation vector of the system at time step k, respectively, as defined in Section III-A. Additionally, let the design undergoing optimization be uniquely defined by a vector of design parameters $\mathbf{p} \in \mathbb{R}^{n_p}$; for example, each component of $\mathbf{p}$ may represent the length of a link along a given CKC (see Section III-B).

Eventually, let the following assumption hold.

Assumption 1. *The initial state $\mathbf{x}_0$ and the vector of design parameters $\mathbf{p}$ satisfy the following system of nonlinear equations:*

$$\mathbf{f}_0(\mathbf{x}_0,\mathbf{p}) = \mathbf{0}_{n_x\times 1}\,, \tag{8}$$

where $\mathbf{f}_0\colon \mathbb{R}^{n_x+n_p} \to \mathbb{R}^{n_x}$ is of class $C^1(\mathbb{R}^{n_x+n_p})$ and is such that the Jacobian matrix $\left.\dfrac{\partial \mathbf{f}_0(\boldsymbol{x}_0,\boldsymbol{p})}{\partial \boldsymbol{x}_0}\right|_{\mathbf{x}_0,\mathbf{p}} \in \mathbb{R}^{n_x\times n_x}$ evaluated at $\langle \mathbf{x}_0,\ \mathbf{p}\rangle$ is full rank.

The purpose of (8) is to provide a well defined initial state given a specific $\mathbf{p}$. For example, $\mathbf{f}_0$ may prescribe the fulfillment of holonomic constraints arising from closed-loop topologies, and subsequently set the remaining unconstrained degrees of freedom to arbitrary values. Most importantly, (8) implicitly redefines the initial state $\mathbf{x}_0$ as a function of $\mathbf{p}$. Indeed, we can employ the result stated in the following theorem [25].

Theorem 1 (Dini's Implicit Function Theorem)**.** *If Assumption 1 is met, then there exists a neighborhood $U_p \subset \mathbb{R}^{n_p}$ of $\mathbf{p}$ and a unique function $\mathbf{x}_0\colon U_p \to \mathbb{R}^{n_x}$ such that $\mathbf{f}_0(\mathbf{x}_0(\mathbf{p}'),\mathbf{p}') = \mathbf{0}_{n_x\times 1}$ and*

$$\left.\frac{\mathrm{d}\mathbf{x}_0(\boldsymbol{p})}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}'} = -\left.\frac{\partial \mathbf{f}_0(\boldsymbol{x}_0,\boldsymbol{p})}{\partial \boldsymbol{x}_0}\right|^{-1}_{\mathbf{x}_0(\mathbf{p}'),\mathbf{p}'} \left.\frac{\partial \mathbf{f}_0(\boldsymbol{x}_0,\boldsymbol{p})}{\partial \boldsymbol{p}}\right|_{\mathbf{x}_0(\mathbf{p}'),\mathbf{p}'} \tag{9}$$

for any $\mathbf{p}' \in U_p$.

For a given vector of design parameters $\mathbf{p}$, we define the initial sensitivity $\mathbf{s}_0(\mathbf{p})$ as the matrix

$$\mathbf{s}_0(\mathbf{p}) := \left.\frac{\mathrm{d}\mathbf{x}_0}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} \in \mathbb{R}^{n_x\times n_p}\,. \tag{10}$$

If a solution of (8) exists for a given $\mathbf{p}$, then the corresponding unique initial state $\mathbf{x}_0 = \mathbf{x}_0(\mathbf{p})$ is a function of $\mathbf{p}$, even though an analytic expression for it may not exist. However, we can always compute the total derivative of $\mathbf{x}_0(\mathbf{p})$ with respect to $\mathbf{p}$ using (9).

Let $\boldsymbol{\varsigma}_k\colon \mathbb{R}^{n_x+n_\tau+n_p} \to \mathbb{R}^{n_x}$ be a differentiable function capturing the simulator dynamics at time step k. Specifically, given $\mathbf{x}_k$, $\boldsymbol{\tau}_k$ and $\mathbf{p}$, we can write $\mathbf{x}_{k+1} = \boldsymbol{\varsigma}_k(\mathbf{x}_k,\boldsymbol{\tau}_k,\mathbf{p})$. Also, let us define the time-varying feedback control policy $\boldsymbol{\mu}_k$ as a differentiable function $\boldsymbol{\mu}_k\colon \mathbb{R}^{n_x+n_p} \to \mathbb{R}^{n_\tau}$ outputting the actuation vector $\boldsymbol{\tau}_k$ at time step k given the corresponding state $\mathbf{x}_k$ and the vector of design parameters $\mathbf{p}$, that is $\boldsymbol{\tau}_k = \boldsymbol{\mu}_k(\mathbf{x}_k,\mathbf{p})$. In this work, $\boldsymbol{\mu}_k$ yields the optimal joint torques as computed by the QP control problem (5).

Finally, the design optimization problem can be formalized as the following nonlinear programming problem:

$$\min_{\mathbf{p}} \; \mathcal{L} := \sum_{k=0}^{N-1} l_k(\mathbf{x}_k, \boldsymbol{\tau}_k) \tag{11a}$$

$$\text{where } \mathbf{f}_0(\mathbf{x}_0, \mathbf{p}) = \mathbf{0}_{n_x \times 1}\,, \tag{11b}$$

$$\mathbf{x}_{k+1} = \varsigma_k(\mathbf{x}_k, \boldsymbol{\tau}_k, \mathbf{p}), \qquad \forall k\,, \tag{11c}$$

$$\boldsymbol{\tau}_k = \boldsymbol{\mu}_k(\mathbf{x}_k, \mathbf{p}), \qquad \forall k\,, \tag{11d}$$

and the terms $l_k \colon \mathbb{R}^{n_x+n_\tau} \to \mathbb{R}$, $k = 0, 1, \ldots, N-1$ are differentiable scalar functions that represent a demerit measure to be minimized.

To optimize the objective function (11a) by using a gradient method, we must evaluate its total derivative with respect to the vector of design parameters $\mathbf{p}$; given the initial sensitivity $\mathbf{s}_0(\mathbf{p})$, this can be done recursively [23]. To this end, we note that the dependency on $\mathbf{p}$ of the initial state $\mathbf{x}_0(\mathbf{p})$ propagates through the constraints (11c) and (11d) to all the input torques and subsequent robot states. Thus, we can write

$$\left.\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} = \sum_{k=0}^{N-1} \left[\left(\frac{\partial l_k}{\partial \boldsymbol{x}_k} + \frac{\partial l_k}{\partial \boldsymbol{\tau}_k} \frac{\partial \boldsymbol{\mu}_k}{\partial \boldsymbol{x}_k} \right) \mathbf{s}_k + \frac{\partial l_k}{\partial \boldsymbol{\tau}_k} \frac{\partial \boldsymbol{\mu}_k}{\partial \boldsymbol{p}} \right], \tag{12}$$

where $\mathbf{s}_k(\mathbf{p}) := \left.\frac{\mathrm{d}\mathbf{x}_k}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} \in \mathbb{R}^{n_x \times n_p}$ is the sensitivity at time step $k = 1, 2, \ldots, N$, and we omitted the dependencies on $\mathbf{x}_k$, $\boldsymbol{\tau}_k$, and $\mathbf{p}$ for the sake of readability.

Given $\mathbf{s}_k(\mathbf{p})$, $\mathbf{s}_{k+1}(\mathbf{p})$ can be recursively calculated for any $k = 0, 1, \ldots, N-1$ by the chain rule as

$$\mathbf{s}_{k+1}(\mathbf{p}) := \left.\frac{\mathrm{d}\mathbf{x}_{k+1}}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} = \frac{\partial \varsigma_k}{\partial \boldsymbol{x}_k} \mathbf{s}_k + \frac{\partial \varsigma_k}{\partial \boldsymbol{\tau}_k} \left.\frac{\mathrm{d}\boldsymbol{\tau}_k}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} + \frac{\partial \varsigma_k}{\partial \boldsymbol{p}}\,,$$

and the terms $\left.\frac{\mathrm{d}\boldsymbol{\tau}_k}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}}$ can be analogously computed as

$$\left.\frac{\mathrm{d}\boldsymbol{\tau}_k}{\mathrm{d}\boldsymbol{p}}\right|_{\mathbf{p}} = \frac{\partial \boldsymbol{\mu}_k}{\partial \boldsymbol{x}_k} \mathbf{s}_k + \frac{\partial \boldsymbol{\mu}_k}{\partial \boldsymbol{p}}\,.$$

V. IMPLEMENTATION

In the following sections, we solve the design optimization problem (11) for *Dog V3*, a dog-like robot with three-segmented hind legs featuring planar CKCs, and we compare the resulting design to *Dog V2*, a quadruped robot with two-link limbs (see Fig. 1). We do so by computing the gradient of the objective function (11a), and then feeding it to the L-BFGS algorithm [26]; our implementation is based on the publicly available LBFGS++ library [27].

A. Experimental Setup

All the results we present were obtained on the same set of control tasks $\mathbb{T}$. Each trial was executed for $4.24\,\mathrm{s}$, which roughly corresponds to 6 strides of the robot trotting gait. We weighted the objective functions by the estimated relative frequency of the respective task in prospective real-world applications; afterwards, we got an overall performance index by summing them up. The different trials are tagged and summarized in Table I. As time step duration h, we used $8.3\,\mathrm{ms}$, which results in $N = 508$ discrete trajectory steps.

TABLE I
CONTROL TASKS

Tag	Description	Weight
STAND	stand at nominal body height (48 cm)	0.17
CROUCHED TROT	trot in place at a body height of 42 cm	0.07
NOMINAL TROT	trot forward at a velocity of 0.4 m/s	0.5
POWER TROT	trot forward at a velocity of 0.8 m/s	0.26

We complemented the QP-based whole-body controller of Section III with inequality constraints of the form (7) bounding the absolute values of the joint torques from above by $60\,\mathrm{N\,m}$. We coupled the controller with a simple kinematic planner that, given the current state of the robot as well as task-specific information, computes the trajectories to be tracked for the robot base and swing feet as Catmull-Rom splines [28]. We calculated the stance foot locations via Raibert heuristics [1, 15].

For the simulations, we performed a forward integration of the equations of motion using a semi-implicit Euler method; i.e., we implemented the simulator dynamics as:

$$\varsigma_k(\mathbf{x}_k, \mathbf{p}) := \begin{bmatrix} \mathbf{I}_{n_q} & h\mathbf{I}_{n_u} \\ \mathbf{0}_{n_q} & \mathbf{I}_{n_u} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} h^2\mathbf{I}_{n_u} \\ h\mathbf{I}_{n_u} \end{bmatrix} \dot{\mathbf{u}}_k^*(\mathbf{x}_k, \mathbf{p})\,, \tag{13}$$

where the vector of the optimal generalized accelerations $\dot{\mathbf{u}}_k^*(\mathbf{x}_k, \mathbf{p})$ was directly output by the whole-body controller.

For the simulation to be stable, we had to modify the CKC constraint formulation (3) within the QP control problem by applying Baumgarte's method as in [24, eq. (2.110)].

We computed the gradient of the objective function (11a) using a double-sided FD method. Although we implemented the SA procedure and assessed its correctness by comparing its outputs to the FD ones, we did not employ it in the optimizations. Indeed, our codebase does not support automatic differentiation; therefore, the modification of any part of the design optimization problem (11) would have required adequate changes in the SA partial derivatives, resulting in an impractical, error-prone process. However, the small number of iterations required by L-BFGS to converge largely compensated for the extra computational cost induced by FD (see Section VI).

To prevent singular configurations from easily occurring in the CKCs [29], we added a relaxed log-barrier term [17, eq. (9)] to (11a) that penalized values for the condition number of the CKC constraint Jacobian matrix above a predefined threshold. We further set the objective function equal to infinity whenever we attained unfeasible closed-loop topologies during the line search phase of L-BFGS. These expedients were sufficient to always ensure the convergence of our approach.

B. Design Performance Metrics

In this section, we present the three performance metrics we used to optimize the hind legs of *Dog V2* and *Dog V3*.

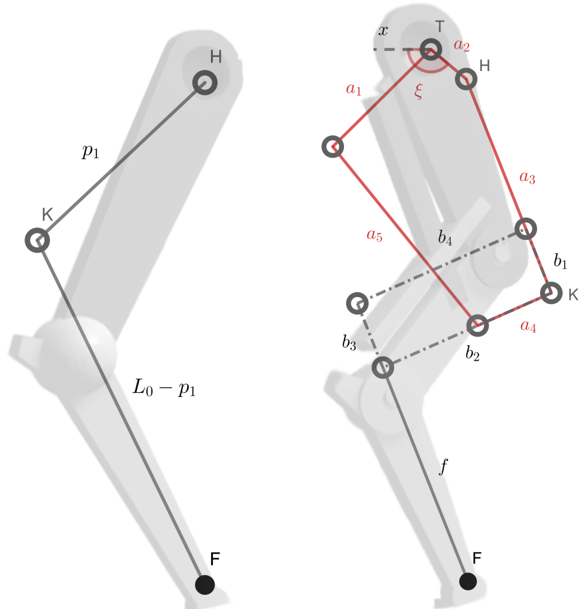

Fig. 4. Leg design diagrams for *Dog V2* and *Dog V3*. The points H and K denote the hip and the knee joints, whereas F denotes the robot's point foot. **Left:** Diagram of *Dog V2*'s hind leg. A motor actuates the knee joint; p_1 denotes the length of the femur, and $L_0 - p_1$ the one of the shank. **Right:** Diagram of *Dog V3*'s hind leg. The only actuated joints are located at T and H. The parameters $a_i,\ i = 1, 2, \ldots, 5$ (resp., $b_i,\ i = 1, 2, \ldots, 4$) denote the link lengths of the upper CKC (resp., lower CKC), which is marked by a red, solid (resp., gray, dashed) line. The pose of the link a_2 is fixed in the two-dimensional leg reference frame; assuming that the position of H is known, we parameterize the orientation of a_2 through the angle ξ with respect to the x-axis[1]. Finally, f denotes the length of the foot.

For each measure, we write the relative objective functions $l_k,\ k = 0, 1, \ldots, N-1$ of (11a).

1) Squared Magnitude of Actuation Vector: We define the average squared magnitude of the actuation vector (SMAV) as the following scalar function [5, 7]:

$$\mathrm{SMAV} \coloneqq \sum_{k=1}^{N-1} l_k^{\mathrm{SMAV}}(\boldsymbol{\tau}_k) = \sum_{k=1}^{N-1} \frac{h}{N} \left\|\boldsymbol{\tau}_k\right\|_2^2 . \tag{14}$$

By minimizing (14), we reduce the strain the actuators are subject to, and thus we cut down the risk of breakage.

2) Power Quality: The power quality quantifies the antagonism in a machine and the balance of power among the motors [30]. It does not take into account the dynamics of the motors, nor any loss model: it solely informs the kinematic design of a mechanism.

The mean power quality (PQ) is defined as:

$$\mathrm{PQ} \coloneqq -\sum_{k=1}^{N-1} l_k^{\mathrm{PQ}}(\mathbf{x}_k, \boldsymbol{\tau}_k) = \sum_{k=1}^{N-1} \frac{h}{N} \left[\left(\mathbf{1}_{1\times n_u} \boldsymbol{\pi}\right)^2 - \left\|\boldsymbol{\pi}\right\|_2^2 \right] , \tag{15}$$

where $\boldsymbol{\pi} = \boldsymbol{\pi}(\mathbf{x}_k, \boldsymbol{\tau}_k) \coloneqq ({}_u\mathbf{S}_x \mathbf{x}_k) \circ (\mathbf{S}^\top \boldsymbol{\tau}_k) \in \mathbb{R}^{n_u}$ is the power tuple of the system at time step k, ${}_u\mathbf{S}_x \in \{0, 1\}^{n_u \times n_x}$ is the selection matrix such that $\mathbf{u}_k = {}_u\mathbf{S}_x \mathbf{x}_k$, and $\circ$ denotes the Hadamard product.

[1]Note that in our discussion about planar CKC constraints of Section III-B, we did not consider the case where one of the links along a CKC is fixed in the mechanism reference frame. However, (3) can easily be extended to this new parameterization.

3) Required Power with Zero-Regeneration Motor Model: We can estimate the energy consumption of a design through a zero-regeneration motor model discarding negative power [30]. In particular, the average power required to perform a task is defined as:

$$\mathrm{PZR} \coloneqq \sum_{k=1}^{N-1} l_k^{\mathrm{PZR}}(\mathbf{x}_k, \boldsymbol{\tau}_k) = \sum_{k=1}^{N-1} \frac{h}{N} \mathbf{1}_{1\times n_u} \left\lfloor \boldsymbol{\pi} \right\rfloor^+ , \tag{16}$$

where the operator $\lfloor \cdot \rfloor^+ : \mathbb{R}^n \to \mathbb{R}^n_{\geq 0}$ sets the negative components of a vector of $\mathbb{R}^n$ to zero.

VI. RESULTS

A. Two-Link Leg

As a preliminary study, we optimized the hind legs of *Dog V2*, a robot dog with no CKCs in its mechanical design (see Fig. 4). In this simple example, we only have one design parameter p_1 denoting the length of the femur link, and we set its initial value to $28\,\mathrm{cm}$.

To compute the various masses and moments of inertia, we model each link as a rigid rod. We additionally model an actuator at the knee joint [7, 12] as a point mass. Finally, we constrain the total length of the stretched leg to be equal to $L_0 = 56\,\mathrm{cm}$ and its total mass to be equal to $M_0 = 2.37\,\mathrm{kg}$.

We optimized the hind leg of *Dog V2* by running L-BFGS, having concentrated 52% of the leg masses in the actuators. The optimal femur lengths corresponding to the SMAV and the PZR objectives are, respectively, $10.08\,\mathrm{cm}$ and $35.84\,\mathrm{cm}$. These results suggest that a short femur and long shank reduce the torque requirements, whereas a long femur and short shank lessen the energy requirements. This trend is in accordance with previous research [5, 7, 9]. However, as was observed by Chadwick et al. [7], two-link designs favor a femur and shank of equal dimension for versatility reasons. Also, most such designs have no motors at the knee joints and employ, e.g., belt transmissions to impart torque signals instead. Thus, in the next section we compare various optimized three-segmented legs to two-link ones with equal link lengths and "massless" actuators.

B. Three-Segmented Leg

Fig. 4 shows a simplified diagram of *Dog V3*'s three-segmented leg. It contains two CKCs and is fully defined by ten parameters, which we stack in a vector $\mathbf{p} \in \mathbb{R}^{n_p} = \mathbb{R}^{10}$. Once again, we constrained the total mass of the leg to be equal to M_0, and we enforced the stretched leg length constraint $L_0 = a_3 + b_2 + f$. For simplicity, we did not adjust the inertial properties of the links as their lengths varied, and we kept them constant instead.

We ran L-BFGS on diverse initial values for $\mathbf{p}$ and using SMAV, PQ, or PZR as objective functions. We additionally optimized for three-segmented designs with coaxial hip and knee actuators; this configuration is found in most existing robots because it is easy to be manufactured and requires no special engineering effort to be implemented. The best outcomes are summarized in Table II, where we included the performance achieved by *Dog V2* with a femur and

TABLE II

RESULTS OF THE OPTIMIZATIONS OF *Dog V2* AND *Dog V3*

In the first column from the left, the name of each optimized design has the form "⟨robot model⟩-⟨optimized measure⟩". We further append the "+" symbol when the corresponding optimization was performed with the parameters a_2 and ξ both fixed to 0 – i.e., the points T and H in Fig. 4 coincided. The second column contains the total number of iterations until L-BFGS converged. The next eleven columns contain the optimized CKC parameters expressed in millimeters or in radians. The last three columns display the values taken on by the performance metrics SMAV [$\mathrm{N}^2\mathrm{m}^2$], PQ [W^2], and PZR [W] in the simulator described in Section V, respectively. If present, the initial values are underlined and written with a light gray font.

Name	Iterations	p_1	a_1	a_2	a_3	a_4	a_5	ξ	b_1	b_2	b_3	b_4	SMAV	PQ	PZR
Dog V2	-	280	-	-	-	-	-	-	-	-	-	-	1183	-427	24
V3-SMAV+	5	-	60 51	-	200 203	60 66	200 197	-	50 42	160 160	50 63	160 160	1287 778	-719 -141	27 21
V3-SMAV	14	-	80 63	40 43	200 195	70 91	200 207	5.9 5.9	60 40	160 162	60 88	160 157	1273 680	-484 -97	24 20
V3-PQ+	11	-	60 53	-	200 204	60 64	200 198	-	50 34	160 160	50 75	160 163	1287 754	-719 -45	27 21
V3-PQ	10	-	70 53	50 34	200 203	70 85	200 198	3.8 3.8	60 41	160 160	60 91	160 159	1745 750	-1594 -141	35 21
V3-PZR+	23	-	70 130	-	200 126	70 188	200 183	-	60 118	160 205	60 264	160 205	1259 658	-685 -53	26 19
V3-PZR	11	-	80 124	40 57	200 167	70 116	200 171	5.9 5.9	60 86	160 149	60 159	160 169	1273 758	-484 -52	24 20

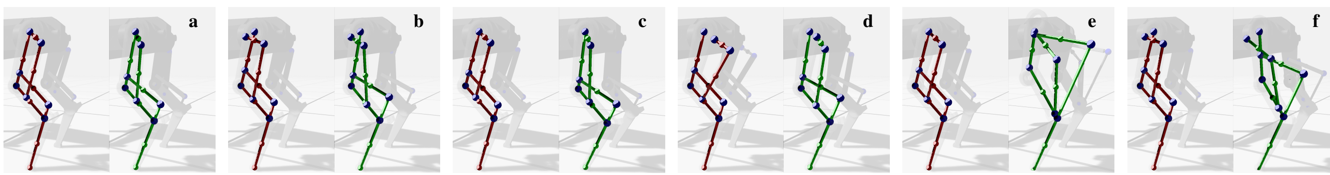

Fig. 5. Initial (left, red) and optimized (right, green) three-segmented leg designs of the robots listed in Table II; namely, *V3-SMAV+* (**a**), *V3-SMAV* (**b**), *V3-PQ+* (**c**), *V3-PQ* (**d**), *V3-PZR+* (**e**), and *V3-PZR* (**f**).

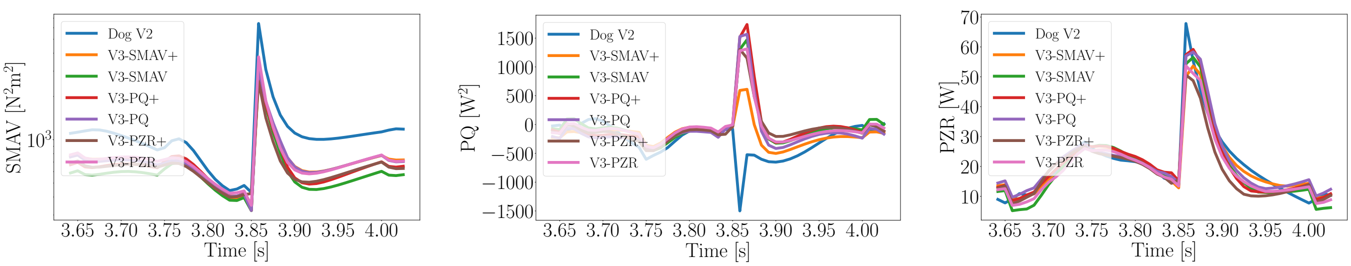

Fig. 6. Performance of the robot designs listed in Table II in the ODE-based simulator on the **NOMINAL TROT** experiment for the SMAV (left), PQ (center), and PZR (right) metrics. To enhance clarity, the y-axis of the leftmost plot is on a logarithmic scale.

shank of equal length. Almost all the optimizations of *Dog V3* converged after less than 15 iterations, despite the large parameter space. On average, each iteration took roughly $2\,\mathrm{min}$ ($n_p = 8$) or $3\,\mathrm{min}$ ($n_p = 10$) due to the gradient calculations by FD. These computation times are competitive to ES- or GA-based approaches. For instance, the smallest optimizations in [7] required about $15\,\mathrm{min}$ to conclude, and the related rollouts did not include a whole-body controller in the loop, as in this work.

The three rightmost columns of Table II allow us to compare the optimized versions of *Dog V3* to *Dog V2*. The results of the former are consistently superior to the latter's by a wide margin. For instance, the performance of *Dog V2* on the SMAV measure is $1183\,\mathrm{N}^2\mathrm{m}^2$, whereas all the results for *Dog V3* exhibit values smaller than $780\,\mathrm{N}^2\mathrm{m}^2$. Identical observations hold for either PQ or PZR.

Although box constraints[2] can be added to steer L-BFGS towards certain directions or to prevent unfeasible designs, we did not enforce any to keep the implementation as simple as possible. Interestingly, the CKCs of *Dog V3* proved well-formed when optimized on either SMAV or PQ, whereas they were much more bizarre when we adopted the PZR measure (see Fig. 5). We assume this comes from the fact that the power requirements of a motion task are intimately related to the motion itself. Thus, it is necessary to resort to exotic solutions to reduce them.

To validate the optimized designs, we evaluated the performance metrics within a different, polished simulator based on the Open Dynamics Engine (ODE) [31]. In this engine, the contacts between rigid bodies undergo both static and dynamic friction, and the system dynamics are integrated stably through an implicit Euler method. In Fig. 6, we show the plots of the SMAV, PQ, and PZR metrics over a single stride for the **NOMINAL TROT** task. All the optimized three-segmented designs consistently outperform *Dog V2* despite the greater complexity of the ODE-based simulator.

VII. DISCUSSION

A. Conclusions

In this work, we leveraged gradient methods to overcome well-known drawbacks of derivative-free design optimizations, such as long optimization times and weak convergence guarantees. We evaluated the performance of a quadruped robot on a set of dynamic motion tasks. We tracked the kinematic plans using a whole-body controller, thereby increasing the optimized designs' awareness of the control ob-

[2]More complicated requirements on the CKC design parameters can be enforced through penalty terms in the objective function [7].

jectives. Finally, by parameterizing general planar CKCs, we could run our method on bio-inspired three-segmented limb architectures featuring such mechanisms. In all our tests, this type of leg achieved conspicuous gains in performance compared to traditional two-link designs. Hopefully, these findings will foster the conception of new breeds of bio-inspired quadruped robots.

B. Limitations and Future Directions

In our experiments, we employed FD to compute the required derivatives for our gradient method. We could significantly improve the time efficiency by employing analytic SA, as it would only cost some extra sparse matrix multiplications at each time step during the forward simulation.

Although we only optimized for the mechanical design of a legged robot, the same methodology can equally be used for any parameter of the controller and the kinematic planner. For this purpose, we may either perform all the optimizations jointly or in an alternate fashion.

For the gradient computations, we utilized a simulator based on a semi-implicit Euler method. Thus, it would be interesting to see what improvements more polished differentiable simulators (e.g., [32]) would bring. Also, by adopting more advanced control algorithms, such as MPC-based strategies, the optimized systems would exhibit even better performances.

Finally, the success of our "sim-to-sim" transfer paves the way for future hardware realizations. We plan to build and test an actual prototype based on the optimized designs presented in this work.

References

[1] M. Raibert, *Legged Robots That Balance*. Cambridge, MA: MIT Press, 1986.

[2] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, pp. 10 822–10 825, 2008.

[3] 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," *2013 IEEE International Conference on Robotics and Automation*, pp. 3307–3312, 2013.

[4] 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, pp. 932 – 950, 2013.

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

[6] ——, "Joint optimization of robot design and motion parameters using the implicit function theorem," in *Robotics: Science and Systems*, 2017.

[7] M. Chadwick, H. Kolvenbach, F. Dubois, H. F. Lau, and M. Hutter, "Vitruvio: An open-source leg design optimization toolbox for walking robots," *IEEE Robotics and Automation Letters*, vol. 5, pp. 6318–6325, 2020.

[8] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "Co-designing robots by differentiating motion solvers," *ArXiv*, vol. abs/2103.04660, 2021.

[9] K. M. Digumarti, C. Gehring, S. Coros, J. Hwangbo, and R. Siegwart, "Concurrent optimization of mechanical design and locomotion control of a legged robot," 2014.

[10] C. Chevallereau, P. Wenger, Y. Aoustin, F. Mercier, N. Delanoue, and P. Lucidarme, "Leg design for biped locomotion with mono-articular and bi-articular linear actuation," *Mechanism and Machine Theory*, vol. 156, p. 104138, 2021.

[11] D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 558–564, 2016.

[12] M. Hutter, C. Gehring, D. Jud, A. Lauber, D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Höpflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016.

[13] D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3359–3365, 2017.

[14] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 4, pp. 3687–3694, 2019.

[15] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *ArXiv*, vol. abs/1909.06586, 2019.

[16] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6295–6301, 2019.

[17] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4730–4737, 2019.

[18] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *ICML*, 2017.

[19] F. Ruppert and A. Badri-Spröwitz, "Series elastic behavior of biarticular muscle-tendon structure in a robotic leg," *Frontiers in Neurorobotics*, vol. 13, 2019.

[20] A. Ananthanarayanan, M. Azadi, and S. Kim, "Towards a bio-inspired leg design for high-speed running." *Bioinspiration biomimetics*, vol. 7 4, p. 046005, 2012.

[21] A. A. Ansari, J. Ahmed, H. Arif, A. Iqbal, H. A. Nizami, and Y. Ayaz, "Investigating series-parallel actuation arrangement in a bioinspired biped robot," *2019 International Conference on Robotics and Automation in Industry (ICRAI)*, pp. 1–5, 2019.

[22] R. Sato, I. Miyamoto, K. Sato, A. Ming, and M. Shimojo, "Development of robot legs inspired by bi-articular muscle-tendon complex of cats," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1552–1557, 2015.

[23] D. Hahn, P. Banzet, J. M. Bern, and S. Coros, "Real2sim: visco-elastic parameter estimation from dynamic motion," *ACM Trans. Graph.*, vol. 38, pp. 236:1–236:13, 2019.

[24] B. Siciliano and O. Khatib, "Springer handbook of robotics," in *Springer Handbooks*, 2016.

[25] M. Bramanti, C. Pagani, and S. Salsa, "Analisi matematica 2," 2009.

[26] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, pp. 773–782, 1980.

[27] Y. Qiu, "Lbfgs++," http://yixuan.cos.name/LBFGSpp, 2016, [Online; accessed 31-December-2020].

[28] E. Catmull and R. Rom, "A class of local interpolating splines," *Computer Aided Geometric Design*, pp. 317–326, 1974.

[29] J. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *Journal of Mechanical Design*, vol. 128, pp. 199–206, 2006.

[30] A. Abate, J. Hurst, and R. L. Hatton, "Mechanical antagonism in legged robots," in *Robotics: Science and Systems*, 2016.

[31] R. Smith, "Open dynamics engine," https://www.ode.org/, 2001-2004, [Online; accessed 31-December-2020].

[32] M. Geilinger, D. Hahn, J. Zehnder, M. Bacher, B. Thomaszewski, and S. Coros, "Add: Analytically differentiable dynamics for multibody systems with frictional contact," *ACM Trans. Graph.*, vol. 39, pp. 190:1–190:15, 2020.