Robotica (2024), **42**, pp. 2004–2025
doi:10.1017/S0263574724000730

RESEARCH ARTICLE

Co-designing versatile quadruped robots for dynamic and energy-efficient motions

Gabriele Fadini[1], Shivesh Kumar[2], Rohit Kumar[2], Thomas Flayols[1], Andrea Del Prete[3], Justin Carpentier[4] and Philippe Souères[1]

[1]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[2]Robotics Innovation Center, DFKI GmbH, Bremen, Germany
[3]Department of Industrial Engineering, University of Trento, Trento, Italy
[4]INRIA and Département d'informatique de l'ENS, Paris, France
Corresponding author: Gabriele Fadini; Email: gfadini@laas.fr

Received: 23 October 2023; **Revised:** 25 March 2024; **Accepted:** 10 April 2024; **First published online:** 9 May 2024

Keywords: legged robots actuation and joint mechanisms methods and tools for robot system design; optimization and optimal control; simulation and animation; mechanism design

Abstract

This paper presents a concurrent optimization approach for the design and motion of a quadruped in order to achieve energy-efficient cyclic behaviors. Computational techniques are applied to improve the development of a novel quadruped prototype. The scale of the robot and its actuators are optimized for energy efficiency considering the complete actuator model including friction, torque, and bandwidth limitations. This method and the optimal bounding trajectories are tested on the first (non-optimized) prototype design iteration showing that our formulation produces a trajectory that (i) can be easily replayed on the real robot and (ii) reduces the power consumption w.r.t. hand-tuned motion heuristics. Power consumption is then optimized for several periodic tasks with co-design. Our results include, but are not limited to, a bounding and backflip task. It appears that, for jumping forward, robots with longer thighs perform better, while, for backflips, longer shanks are better suited. To explore the tradeoff between these different designs, a Pareto set is constructed to guide the next iteration of the prototype. On this set, we find a new design, which will be produced in future work, showing an improvement of at least 52% for each separate task.

1. Introduction

Quadruped robots are becoming of widespread use for practical applications and are starting to be commercially available for automated task [1]. These platforms show their promise in security, patrolling, monitoring, and inspection (e.g., in secluded sites such as off-shore platforms [2]). Quadrupeds are ideal for these uses, thanks to their increased locomotion capability. However, system designers have to face numerous challenges when creating a new robotic platform. Given the complexity of legged robots, it is not trivial to predict how to select the best platform to perform a given set of tasks. This is even exacerbated by the fact that design and control are usually considered separately, while in reality, they are deeply interconnected. Splitting them into subsequent phases leads to an inefficient process in which the design is modified and tested multiple times before reaching an adequate performance and can lead to sub-optimal results. To exploit the system properties at best, the optimization of the robot hardware for the task is hence needed. This concurrent optimization approach takes the name of co-design.

Numerous highly dynamic quadrupedal designs, including both commercial and research platforms, have been developed in the last decade. The most notable ones include Spot [3] by Boston Dynamics, ANYmal [4] from ETH Zürich, the Cheetah series [5–7] from MIT, and the Unitree lineup [8]. Especially after the seminal work on the open-source MIT mini Cheetah robot [9], which has demonstrated back flips, and other highly athletic behaviors, various other smaller-sized quadrupedal platforms became

Table I. *Comparison of some state-of-the-art quadrupeds in terms of their dimensions and dynamic capabilities.*

Quadruped	**Spot [3]**	**ANYmal [4]**	**MIT Mini Cheetah [9]**	**Unitree A1 [8]**
Length (m)	1.1	0.8	0.48	0.5
Width (m)	0.5	0.6	0.27	0.3
Height (m)	0.61	0.7	0.3	0.4
Weight (kg)	32.7	30	9.0	12.7
Battery (Wh)	605	650	NA	90.72
Max speed (m/s)	1.6	1.0	2.45	3.3
Standard tairss†	Yes	Yes	No*	No*
Backflip	No	No	Yes	Yes

*No public demonstration of this skill to the authors' best knowledge.
†Standard stairs with step height = 0.19 m, depth = 0.26 m.

popular. Table I compares some selected quadrupeds' physical dimensions and dynamic capabilities, including maximum speed for walking/running, ability to climb standard stairs and perform a backflip. It can be observed that the smaller-sized quadrupeds are capable of more athletic behaviors (e.g., running with higher speed and the ability to perform a backflip). On the other hand, quadrupeds with larger body lengths can be deployed in real-world environments developed for human accessibility (e.g., climb stairs). A natural question arises: how can we design quadruped robots that can optimally perform in a range of movements?

Co-design has been historically first used to optimize the motion together with the controller, for instance, with the discovery of the optimal trajectory with the associated gains in refs. [10, 11]. On one side, model-based paradigms for legged systems hardware selection, featuring several design criteria, have been proposed in refs. [12–15]. On the other side, optimal control was used to optimize the motion of biped together with its kinematic parameters to produce stable running by using local trajectory optimization coupled with genetic optimization for the hardware parameters [16]. Several contributions already dealt with the problem of robot co-design combining both hardware and control selection. In ref. [17, 18], passive walker actuators were optimized for cyclic motions. Design kinematic parameters were chosen in order to produce smooth motion for mechanical avatars in ref. [19], while in ref. [20] the leg design of the StarlETH was selected to optimize peak speed. In ref. [21] several simple-legged robots were designed in a single NLP problem where the hardware was optimized at the same level of the motion. Hard constraints on task fulfillment could be enforced in the problem. Other work focused on the optimality of motion and design, for instance in refs. [22, 23] monopeds were designed to minimize different cost functions, targeting energy efficiency. In ref. [24], robot designs were optimized to follow user-defined trajectories changing just the link scaling of the legs. The method exploits the implicit function theorem to obtain a manifold of feasible solutions in the design space. More recently, in ref. [25], a framework to optimize legged robot design in order to track trajectories planned with the single rigid body dynamic assumption was introduced. The advantage of this framework is the possibility to change freely the metrics to generate different designs. However, this is at the loss of the optimality of pre-selected trajectories generated by a simplified motion planner that cannot fully exploit the system dynamics. In ref. [26] a co-optimization algorithm is also presented for the quadruped Solo. Differentiation of the motion planner is exploited in order to obtain faster convergence and impose arbitrary constraints on the design variables. Tackling the problem of robustness in co-design has been the focus of refs. [27, 28] where stochastic optimization was used to improve the performance of the system under unplanned perturbations. An extension was proposed in ref. [29], where to make the stochastic approach scalable, an ADMM method is used to optimize the robot design with the main goal to increase control robustness with respect to different scenarios. The results feature the optimization of a planar quadruped bounding gait for the mini Cheetah robot [30]. Some preliminary results in integrating the trajectory stabilization at the design level for simple underactuated systems can be found in

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

ref. [31, 32]. In refs. [33, 34], two successive works were conducted to develop a generic framework to cover legged robots co-design combining trajectory optimization and genetic algorithms. Additional experimental work validating hardware selection choices was performed in ref. [35]. Among these contributions, only a few works really achieved developing a general framework for co-design and drawing the link with real hardware implementation and testing. This is the objective of the current work.

Contributions. In this paper, we present an extension of the co-design framework introduced in refs. [33, 34] in order to make it more complete and versatile, and we apply it to improve the design of a new quadruped robot developed at the Underactuated Robotics Lab of DFKI-RIC in Bremen. The key contributions are as follows:

- Development of a more computationally robust optimal control problem formulation without increasing the overall computation times (compared to our previous work [33, 34], parallelization is employed to cope with the higher complexity of the platform and tasks to optimize. We remark that robustness here is referred to the trajectory optimization algorithm and that this work is not dealing with robust co-design).
- Accurate modeling of the actuators (critical for co-design), which features a re-formulation of bandwidth limitation as path constraints. Our energy and actuator models used in trajectory optimization are also validated on the bounding task, showing that they can be replayed with increased energy efficiency (w.r.t. hand-tuned trajectories that achieve similar displacement).
- Optimization of several cyclic movement patterns by enforcing non-Markov constraints between the final and initial state, which are discovered automatically. In practice, we synthesize but are not limited to, bounding and backflip optimal motions. The performance in realizing these behaviors is then considered in our co-design framework as different goals to achieve.

The paper is organized as follows. The rationale and theoretical aspects of the framework are outlined in Section 2. In Section 3, we focus on the optimal control problem formulation. Here the actuator model is presented together with its impact on the constraints (actuator and bandwidth limitation) and cost function (electrical power consumption). In Section 4.1, the current quadruped development at DFKI is described. The actuator model is then proven to provide good estimates on the current hardware implementation through experimental validation. Later, a co-design study on the platform is performed, and the results are collected in Section 5. In particular, in Sections 5.2 and 5.3, the designs are preliminary optimized for a single task (respectively, bounding, and backflips). Then, to select an improved design of the platform, a refinement that considers both tasks is shown in Section 5.4.

2. Methodology

2.1. Co-design framework structure and characteristics

Figure 1 depicts our parallelized co-design algorithm. Our method relies on a bi-level scheme. In the outer loop, a genetic algorithm optimizes the design parameters considering their optimal cost value $\mathcal{L}$ obtained in the inner loop (a trajectory optimization). The outer loop generates a population of random designs, and for each design, a task-driven optimal control problem (OCP) is solved. After all the individuals of the population are evaluated, the outer loop proceeds with the evolution of the population, generating a new random population propagating the information of the designs that provided the best cost.

Outer loop (genetic algorithm)

To optimize over the hardware parameter space, we use a gradient-free, population-based stochastic optimization CMA-ES [36]. This makes the method less impacted by the presence of local minima. This is not generally true in the case of gradient-based co-design approaches [24], which depend on an initial guess. One advantage of genetic algorithms (GA) is that they can work with integer variables. Given our

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

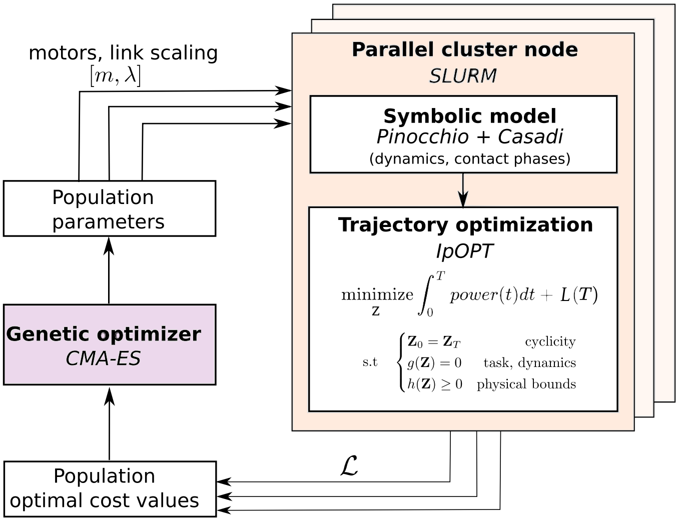

motors, link scaling
$[m, \lambda]$
Parallel cluster node
SLURM
Symbolic model
Pinocchio + Casadi
(dynamics, contact phases)
Trajectory optimization
IpOPT
$\underset{\mathbf{Z}}{\text{minimize}} \int_0^T power(t)dt + L(T)$
s.t $\begin{cases} \mathbf{Z}_0 = \mathbf{Z}_T & \text{cyclicity} \\ g(\mathbf{Z}) = 0 & \text{task, dynamics} \\ h(\mathbf{Z}) \geq 0 & \text{physical bounds} \end{cases}$
Population parameters
Genetic optimizer
CMA-ES
Population optimal cost values
$\mathcal{L}$

Figure 1. ***Overview of the approach****. Stack of the parallelized bi-level optimization scheme with arbitrary hard constraints on the primal optimization variables.*

problem size, we propose a strategy to optimize discrete motor selection with the other design variables via CMA-ES. Finally, the genetic approach in the outer loop is *massively parallelizable*. Thanks to this property, the overall computation time is reduced, as the whole optimization framework was adapted for parallelization on a high-performance cluster (HPC), using *SLURM* in Fig. 1.

Inner loop (OCP solver)

Guarantees on task fulfillment are enforced by hard constraints, which are now supported by using the state-of-the-art interior point solver *IpOPT* [37] for solving the OCP. In our previous work [33, 34], strict equality and inequality constraints could not be exactly enforced, but only approximated by penalties in the cost function. This was rather limiting because it required hand-tuning the weights and parameters associated with the penalties. Such time-consuming and error-prone process was a main source of brittleness, which we have overcome in this work by relying on *Casadi* [38] and *Pinocchio* [39, 40]. Now more versatile, yet complex, optimal control problem formulations can be solved with robust general-purpose optimizers. *IpOPT* comes with a robust optimization routine that allows a better globalization compared to other state-of-the-art gradient-based solutions. However, this comes at the expense of:

- increased computation time compared to iLQR or DDP [41], as the specific sparsity pattern of the OCP is not exploited. Moreover, the addition of inequality and equality constraints increases the NLP complexity.
- warmstart capability; because of the barrier initialization, interior point methods are more difficult to warm start. This prevents the reuse of previously computed solutions to solve a new problem instance [42, 43].

This study is limited to the case of trajectories with pre-specified phases and timings (the sequence of contacts is fixed a priori). We apply in co-design a holistic approach inspired by ref. [44] (and later [16, 45]) where the motion of a biped is synthesized by imposing periodic constraints on the trajectory. Similarly to ref. [46], where, without considering the design, the joint trajectories of a planar biped are optimized, cyclic behaviors are imposed in our work through contact constraints and joint limits. Table II compares our method with other state-of-the-art co-design strategies. The following aspects of the method are considered: the aim of the optimization (if the goal is *energy optimization* or to reject perturbations through *robust co-design*), the capability of the method to *explore globally* the design

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Table II. *Comparison between various state-of-the-art co-design approaches.*

Contribution	**[25]**	**[24]**	**[21]**	**[16]**	**[20]**	**[22, 23]**	**[27, 28]**	**[29]**	**[26]**	**[33]**	**[34]**	***This***
System scaling	√	√	√	√	√	✗	√	√	√	√	√	√
Full system dynamics[⋆]	✗	✗	✗	√	√	√	√	√	(√)	√	√	√
Energy optimization	√	✗	✗	✗	✗	√	√	√	√	√	√	√
Hard constraints	✗	✗	✗	√	√	√	√	√	✗	✗	✗	√
Cyclicity	✗	✗	✗	√	✗	√	✗	√	✗	✗	✗	√
Robust co-design[†]	✗	✗	✗	✗	✗	✗	√	√	✗	✗	√	✗
Control policy	✗	✗	✗	✗	✗	✗	√	√	√	✗	√	✗
Global exploration	✗	✗	✗	√	✗	✗	✗	✗	✗	√	√	√
Actuator bandwidth	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	√
Mixed integer variables	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	√
Experimental results	✗	✗	√	✗	√	✗	✗	✗	✗	✗	✗	√

[⋆] Exact dynamics instead of kinematic/reduced models.
[†] Robustness to unpredicted perturbations.

space, the model used in the OCP (especially if the *full system dynamics* is considered instead of simplified models, and the capability to take into account the *actuator bandwidth* constraints), the co-design output (for instance the *system scaling*,. "that is, link dimension or the optimal *control policy*), the possibility of ensuring *hard constraints* or *cyclic ones* and the capability to include *mixed integer variables*. Finally, we also consider the presence of *experimental results* on hardware prototypes. Concerning these different methods, it can be noticed that the work that is most similar to ours in terms of optimized platform and trajectories is ref. [29]. The main advantage of our method is the automatic discovery of the footholds, as the contact location is left free. It is nonetheless possible to further refine the phase timing with black-box techniques as in ref. [33] or with methods that optimize the length of each contact phase in the optimal control problem directly [16]. The strength of our approach is the capability to accurately reflect actuator limitations and performance in complex tasks. Moreover, thanks to the use of GA our versatile bi-level framework can explore globally the discrete optimization variables, which may be problematic in gradient-based methods (which are more impacted by initialization and require a continuous problem structure).

3. Trajectory Optimization Problem

3.1. Formulation and variables

Numerical trajectory optimization is a powerful and versatile tool for robotics. Optimizing a tailored cost function allows to generate a control trajectory for the robot so that it performs specific behaviors [47–50]. The main advantage of this approach is the intuitiveness of setting the cost and constraints, which are strictly related to high-level goals that must be fulfilled. For the trajectory optimization problem on a discretized horizon with nodes $[0..N]$, we use direct collocation with an augmented set of variables: $\mathbf{Z} = [\mathbf{X}, \mathbf{U}, \mathbf{A}, \mathbf{F}, \mathbf{\Gamma}]$ where:

- $\mathbf{X}$ is the decision vector collecting the evaluations of states of the robot $\mathbf{x}$, each state includes the configuration and velocity of all its degrees of freedom. where $\mathbf{q}_{\mathrm{b}}$ is the underactuated base position and $\mathbf{q}_{\mathrm{a}}$ is the vector of actuated joint positions.

$$\mathbf{x} = [\,\underbrace{x, z, \theta, q_{1..n_u}}_{\mathbf{q}=[\mathbf{q}_{\mathrm{b}},\mathbf{q}_{\mathrm{a}}]\in\mathbb{R}^{n_q}}\,\underbrace{\dot{x}, \dot{z}, \dot{\theta}, \dot{q}_{1..n_u}}_{\mathbf{v}=[\dot{\mathbf{q}}_{\mathrm{b}},\mathbf{v}_{\mathrm{a}}]\in\mathbb{R}^{n_v}}\,] \in \mathbb{R}^{(n_x=n_q+n_v)} \tag{1}$$

- $\mathbf{U}$ contains the actuated joint torques $\mathbf{u} \in \mathbb{R}^{n_u}$.
- $\mathbf{A}$ is the vector of the joint accelerations $\mathbf{a} = \dot{\mathbf{v}} \in \mathbb{R}^{n_v}$.

Just for the contact phase nodes $C = [N_{c,0}..N_{c,T}] \subseteq [0..N]$, the foot position $\mathbf{p}_{c,f}$ for the feet f in contact is fixed. For any foot f in contact, we define additionally:

- **F**: contact force vector, which stacks the contact forces on the contact f for all the nodes in which a given contact phase C is defined, $\mathbf{f} = \bigcup_C \bigcup_f (\mathbf{f}_{c,f} \in \mathbb{R}^{n_c})$, where n_c is the contact point dimension, which depends on the contact model. For instance, in planar models $n_c = 2$, while for three-dimensional and contact wrench models it equals $n_c = 3$ and $n_c = 6$, respectively.
- **Γ**: slack variable vector, which collects the contact slack variables acting on the contact f for all the nodes in which a given contact phase C is defined, $\boldsymbol{\gamma} = \bigcup_C \bigcup_f (\boldsymbol{\gamma}_f \in \mathbb{R}^{n_c})$, following the formulation in ref. [48], to impose constraints on the feet position and velocity. These variables are introduced only in the contact phases to avoid contact drift.

The choice of considering such an augmented set of variables, at torque and acceleration level, is motivated by the intention of imposing constraints which reflect the physical limitations of the actuator. This is not directly possible in the case of simplified models such as linear inverted pendulum [51, 52], spring-loaded inverted pendulum [53], or centroidal model [54, 55], which do not include the joint torques. Finally, highly dynamic behaviors are difficult to discover as they are often far from the simplified model assumptions.

3.2. *Optimal control problem constraints*

With the formulation outlined in Section 3.1, constraints can be imposed directly on the primal variables both in the form of equality and inequality constraints. This is an aspect of the utmost importance for co-design, as the feasibility of the motion needs to be guaranteed from the optimization stage.

Robot dynamics.

The robot state $\mathbf{x}$ evolves under the influence of the joint torques and contact forces as described by the constrained rigid body dynamics [56]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ -\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{u} - \mathbf{b} \\ -\dot{\mathbf{J}}_c \mathbf{v} \end{bmatrix}, \tag{2}$$

where $\mathbf{M}$ is the joint space inertia matrix, $\mathbf{b}$ is the vector containing the state-dependent nonlinear effects of gravity, centrifugal and Coriolis forces, and $\mathbf{J}_c$ is the contact Jacobian stacking the Jacobians of all the contact points. Based on this dynamics, the robot configuration $\mathbf{q}$ and its velocity $\mathbf{v}$ evolve under the control action $\mathbf{u}$ of the motors. Equation (2) can be solved using the forward dynamics [56], leading to a constraint on $\mathbf{a}$. Joint accelerations must then be integrated numerically to obtain joint velocities and positions. To this aim, we introduce an integration function $\boldsymbol{\Phi}$, which we used to formulate the following constraints

$$\mathbf{x}^+ = \boldsymbol{\Phi}(\mathbf{x}, \mathbf{a}, \mathbf{u}, \underbrace{\mathbf{f}, \boldsymbol{\gamma}}_{\text{if contact}}, \Delta t) \tag{3}$$

To improve numerical conditioning, the contact point velocity is corrected with a slack variable $\boldsymbol{\gamma}$ as proposed in ref. [48]. This allows redundant constraints on the contact location and its velocity, avoiding drift. This modification is propagated in the integrator law (3), as shown in Eq. (13) of ref. [48]. In the cost function, these slack variables are penalized for converging to physically accurate solution. Our integrator hence depends also on γ because, instead of the state velocity $\mathbf{v}$, the value $\tilde{\mathbf{v}}$ is used in the integration step, where $\tilde{\mathbf{v}}$ can be interpreted as the velocity projected in the kernel space of the contact velocity $\mathbf{J}_c\mathbf{v}$:

$$\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{J}_c^\top \boldsymbol{\gamma} \tag{4}$$

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Contact constraints.

The rigid contact model leads to several constraints described in the following.

Forces. The non-sliding and unilaterality conditions impose the following constraints on any contact force (for flat ground) $\mathbf{f}_c = [f_{c,x}, f_{c,y}, f_{c,z}]^\top$, given the friction coefficient μ:

$$\begin{cases} \mu^2 f_{c,z}^2 \geq f_{c,x}^2 + f_{c,y}^2 \\ f_{c,z} \geq 0 \end{cases} \tag{5}$$

Non-sliding contact points. During any contact phase of horizon $C = [N_{c,0}..N_{c,T}] \subseteq [0..N]$, the position $\mathbf{p}_{c,f}$ of any foot f in contact is constant for the whole phase. In particular, we set it equal to the value at the beginning of the phase:

$$\mathbf{p}_{c,f}(\mathbf{q}_i) = \mathbf{p}_{c,f}(\mathbf{q}_{N_{c,0}}), \quad \forall\, i \in C, i \neq N_{c,0} \tag{6}$$

Non-penetration. The z coordinate of the contact point must be at ground level (flat ground assumption):

$$\mathbf{p}_{c,f}(\mathbf{q}_{N_{c,0}})|_z = 0 \tag{7}$$

Because of (6), this condition can be imposed just on the initial contact node $N_{c,0}$.

Contact velocity. The velocity of the feet in contact must be zero:

$$\mathbf{v}_{c,f}(\mathbf{q}_i) = 0, \quad \forall\, i \in C \tag{8}$$

Collision avoidance with the ground.

To produce a feasible motion, constraints on the vertical position of some key-frames (e.g., shoulder and knee joints, indicated with the subscript k) need to be imposed in order to not penetrate the ground. This is enforced along the whole optimization horizon through inequalities of the type:

$$\mathbf{p}_k(\mathbf{q}_i)|_z \geq 0, \quad \forall\, i \in [0..N] \tag{9}$$

Cyclicity.

As noted in ref. [57], limit cycles naturally arise from trajectory optimization in the simple cases of optimal locomotion. Cyclic motion patterns are then our chosen optimization target to obtain energy-efficient hardware. This choice also makes it possible to keep the optimization horizon per cycle short and therefore reduces computational complexity without sacrificing numerical precision. Once the motion primitive is obtained, a locomotion pattern that is representative of the robot's main operation can be achieved by replicating the cycle multiple times. The periodicity of the solution is introduced in the OCP with non-Markovian constraints between the optimization variables at the initial and final nodes of the problem. Depending on the problem requirements, these constraints can involve the full set of decision variables $\mathbf{z}$ or just a subset of it.

$$\mathbf{g}(\mathbf{z}_0, \mathbf{z}_N) = 0 \tag{10}$$

For instance, some offsets or inequalities can be introduced just on specific parts of the state to enforce a given behavior (e.g., in a forward jump, we want that the base position translates at least of a given amount, but all the other variables match the values at the beginning of the trajectory).

$$\mathbf{h}(\mathbf{z}_0, \mathbf{z}_N) \leq 0 \tag{11}$$

These constraints enforce a dependency between the initial and final nodes. The major drawback is that the requirements to define a Markov chain are not respected anymore. This breaks the NLP sparsity pattern and makes faster iterative algorithms as DDP [41] not viable.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Actuator model and limits.

The choice of the motor is modeled so to change cost function (power consumption) and limits in a physical consistent way. All the main actuator limits are taken into account as a set of constraints:

- *Position*: joint position bounds are considered
$$\underline{\mathbf{q}} \leq \mathbf{q}_a \leq \overline{\mathbf{q}} \tag{12}$$
where $\mathbf{q}_a$ are the actuated joints angular positions and $\underline{\mathbf{q}}, \overline{\mathbf{q}}$ are their minimum and maximum allowable values (e.g., the knee joint angle is delimited by the presence of stoppers).
- *Velocity*: each actuator speed limit is considered by imposing bounds on the joint angular velocity.
$$\underline{\mathbf{v}} \leq \mathbf{v}_a \leq \overline{\mathbf{v}} \tag{13}$$
where $\mathbf{v}_a$ are the actuated joints angular velocities and $\underline{\mathbf{v}}, \overline{\mathbf{v}}$ are their and maximum allowable values. For highly dynamic trajectories, this aspect is essential as these thresholds may easily be reached.
- *Torque*: Generally, torque limits are modeled as fixed bounds on $\mathbf{u}$. This is a necessary but not sufficient condition because the actuator cannot instantly provide arbitrary torque values: the intrinsic limitation due to the bandwidth of the actuation needs to be addressed. Approaches to treating it were proposed in refs. [58, 59] working in the frequency domain respectively on the cost function and to obtain feedback gains that can be applied to the real system. Our approach is to impose physically driven bounds on the torque values themselves. The rationale is that, considering the joint transmission, the elastic elements (particularly the transmission belt) can store energy through small deformations. This acts as a low-pass filter from the motor to the connected joint, which can be approximated by a first-order filter whose cutoff frequency depends on the actuator technology. In time domain, the filter presents a straightforward implementation. For each node $k \in [1..N]$ it results in the following constraints:
$$\begin{cases} \mathbf{u}_k \leq (1-\alpha)\mathbf{u}_{k-1} + \alpha\overline{\mathbf{u}} \\ \mathbf{u}_k \geq (1-\alpha)\mathbf{u}_{k-1} + \alpha\underline{\mathbf{u}} \\ \underline{\mathbf{u}} \leq \mathbf{u}_0 \leq \overline{\mathbf{u}} \end{cases}, \tag{14}$$
where $\alpha \in [0, 1]$ depends on f_c and the discretization step Δt as follows:
$$\alpha = \frac{2\pi\,\Delta t f_c}{2\pi\,\Delta t f_c + 1}. \tag{15}$$
$\underline{\mathbf{u}}, \overline{\mathbf{u}}$ are respectively the minimum and maximum torque that can be achieved by the actuator. By construction, (14) respects peak limits, as $\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \overline{\mathbf{u}}\ \forall k \in [1..N]$.

3.3. *Power cost function*

For what concerns the cost function that is minimized, in the Lagrange term, the total electrical energy consumption is included as the time integral $\int_0^T P_{el}(t)dt$ of electrical power P_{el}, as in refs. [33, 34]. P_{el} is computed with the non-ideal dissipations of the actuators.

Joint friction. The power dissipation due to friction is computed from the identified values of static friction $\boldsymbol{\tau}_\mu$ and viscous friction b.

$$P_f = \mathbf{v}_a^\top \underbrace{(\tau_\mu \text{sign}(\mathbf{v}_a) + b\mathbf{v}_a)}_{\tau_f}, \tag{16}$$

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

where $\mathbf{v}_a$ is the velocity of the actuated joints. We denote as

$$\boldsymbol{\tau} = \mathbf{u} + \boldsymbol{\tau}_f \tag{17}$$

the total joint torque including the friction component $\boldsymbol{\tau}_f$.

Joule effect. The Joule power losses are included on the motor side with the values of the motor constant K coming from its specifications:

$$P_t = \boldsymbol{\tau}^\top \mathbf{K} \boldsymbol{\tau}, \tag{18}$$

where $\mathbf{K}$ is a diagonal matrix containing, for each joint, the reciprocal of the motor constant divided by its squared gear ratio.

Mechanical power. The mechanical power flow to/from the system can be obtained by considering the instantaneous power at the level of the actuator:

$$P_m = \mathbf{u}^\top \mathbf{v}_a \tag{19}$$

Total electrical power. The total electrical power is equal to the losses of the system plus the mechanical power:

$$P_e = P_f + P_t + P_m, \tag{20}$$

Mechanical energy invariance. For any periodic trajectory Ω, parametrized by the variable ξ, the electrical energy equals the integral of the losses $(P_f + P_t)$. Therefore, it is not necessary to minimize explicitly the mechanical power P_m because its circuitation is a conserved quantity and equals the difference in mechanical energy between the final and initial state (which is state-dependent and hence zero by definition of periodicity):

$$\oint_\Omega \mathbf{u}^\top(\xi)\mathbf{v}_a(\xi)d\xi = (\mathrm{E}_{mec} = \mathrm{E}_{kin} + \mathrm{E}_{pot})|_{\mathbf{x}_0}^{\mathbf{x}_T=\mathbf{x}_0} \triangleq 0 \tag{21}$$

This result can also be extended for semi-periodic trajectories. In particular, we consider the case in which joint velocities are the same and only the x position of the robot base changes. Any translation of the base along x is tolerated as it results in no net change in potential energy because:

- the base lands at the same z position it started from
- the actuated joint position trajectories are cyclic

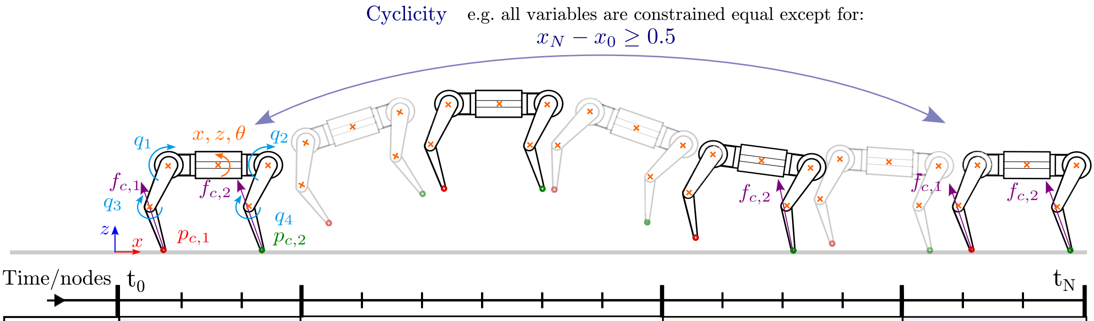
This is a sufficient condition: the final height of each link CoM is equal to the initial one, so no difference in potential energy is induced, and kinetic energy is conserved as there is no difference in state velocity (and the joint space inertia is invariant to base translations).

3.4. Problem specification example

Before presenting the optimization cases, for clarity, the notions introduced in Section 3 are visually represented and discussed in Fig. 2 with a toy problem involving a planar model. In such an example, the goal is to obtain a jump of at least 0.5 m. The specification of the task starts from the selection of several contact phases and timing. To perform this task, 4 distinct phases are chosen: (i) at first, the robot is in contact with both feet (rear and front, *double support*), then (ii) initiates the jump with both feet (*flying phase*), (iii) lands on the front legs (*single support*), and finally (iv) reaches double contact with the ground again (*double support*). In Fig. 2, the different phases are shown with different coloring, and for each of them, the node variables are reported together with the constraints and cost functions.

It is worth mentioning that some variables, costs, and constraints are phase-dependent. For instance, in the flying phase, the slack variables $\boldsymbol{\gamma}$ and the contact forces $\mathbf{f}_c$ are not necessary, so they are not

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press



Phase	Double support	Flying phase	Single support	Double support
Variables	$z_0 = \left\{ \begin{matrix} x_0, z_0, \theta_0, q_a \\ v, a, u, f_c, \gamma \end{matrix} \right\}$	$\left\{ \begin{matrix} x, z, \theta, q_a \\ v, a, u \end{matrix} \right\}$	$\left\{ \begin{matrix} x, z, \theta, q_a \\ v, a, u, f_c, \gamma \end{matrix} \right\}$	$z_N = \left\{ \begin{matrix} x_N, z_N, \theta_N, q_a \\ v, a, u, f_c, \gamma \end{matrix} \right\}$
Constraints	Contacts ●● (6) (7) (8) Collision avoidance × (9) Dynamics (3) Actuators (12) (13) (14) Friction cones (5)	Collision avoidance ●●×(9) Dynamics (3) Actuators (12) (13) (14)	Contact ● (6) (7) (8) Collision avoidance ●×(9) Dynamics (3) Actuators (12) (13) (14) Friction cones (5)	Contacts ●● (6) (7) (8) Collision avoidance × (9) Dynamics (3) Actuators (12) (13) (14) Friction cones (5)
Costs	Electrical power (20) Slack variables $\gamma^\top\gamma$	Electrical power (20)	Electrical power (20) Slack variables $\gamma^\top\gamma$	Electrical power (20) Slack variables $\gamma^\top\gamma$

Figure 2. *Example of a problem construction for a cyclic task of forward jumping of at least 0.5 m.*

introduced. The same goes also for friction constraints, contact constraints, and slack variable penalization (as $\boldsymbol{\gamma}^\top\boldsymbol{\gamma}$). Similarly, also the penalization of the slack variables is only present when the slack variables are defined, that is only in the contact phases. The method consistently provides results with $|\boldsymbol{\gamma}| < 10^{-6}$, when the penalization weight is 10^5. The only other cost component we introduce is related to energy consumption, which has a relatively low weight of 10^{-3}. This cost is present in all phases and is the main target of the optimization. Concerning the constraints, the dynamics constraint, the actuator limits, and the collision avoidance are imposed on all nodes, as can be seen by the row "Constraints." The jumping task is enforced through an inequality in the x base position among the initial and final nodes of the trajectory, in this example $x_N - x_0 \geq 0.5$, while all the other optimization variables are constrained to be equal between initial and final nodes. The last constraint is non-Markovian and overarches the whole optimization problem. Starting from this simple example, similar contact and cyclic constraints allow to define also the problems which we present and solve in the results part. The strength of this problem formulation is that it can generate very different movements with just the selection of the final constraints and gait pattern as will be shown in the following results with bounding and backflip. Those will still be made up of the same OCP formulation that we have here defined. The different possible motions which are achievable can be seen also in the companion video.

4. Real Hardware Results

In this section, we present the current quadruped prototype built at DFKI, and we use it for validating the actuator and the power consumption models introduced in Section 3.3. More precisely, our goal is to validate the hardware model used in trajectory optimization (repeatability of the motion on the real system and the accuracy of the power consumption estimation). Experimental validation of the optimized hardware is not the topic of this section. It will be the subject of future work.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

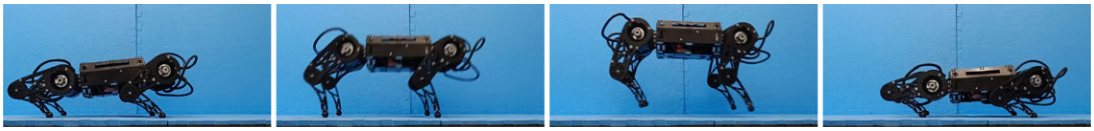


Figure 3. ***Quadruped prototype*** *bounding tests at DFKI-RIC*[1].

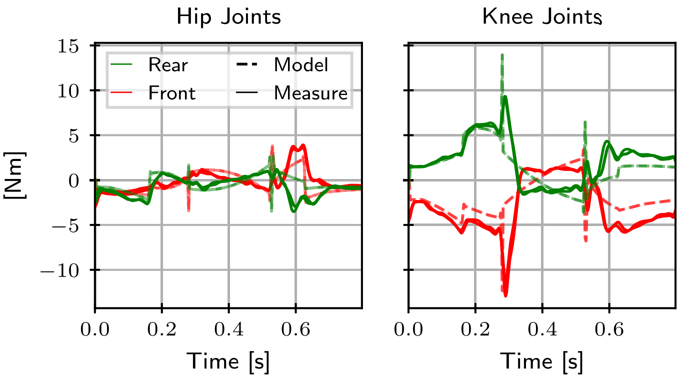


Figure 4. *The actuator model allows a close match between the ideal trajectories with friction compensation and the ideal torque applied to the system from measurement data.*

4.1. The new quadruped prototype at DFKI

The DFKI Robotics Innovation Center recently developed a new robot quadruped (see Fig. 3). The physical dimensions for the initial design are similar to the mjbots-quad [60]. The validation and optimization results are based on the preliminary design, which is presented in this section. The goal of such research platform is to be capable of producing dynamic motions while keeping a certain degree of autonomy. The robot consists of a central body on which four legs of identical design are mounted. Each leg has 3 degrees of freedom (DoF), which are actuated by off-the-shelf quasi-direct drive actuators based on open-sourced MIT mini Cheetah actuators. The hip joint has one pitch, and one roll DoF and the knee joint can be rotated around a pitch axis. To keep the leg's inertia low, the knee joint's motor was shifted to the pitch axis of the hip joint and coupled to the knee joint via a toothed belt transmission with a ratio of $\frac{1}{2}$. All structural elements of the robot were designed in such a way that they can be manufactured by waterjet cutting (including the gearbox pulleys). The components of the body consist of carbon fiber-reinforced plastic (FRP) plates with a thickness of 1 mm. This allows easier manufacturability and assembly, without sacrificing rigidity and lightness. The connections between the hip drives were made from 3 mm thick carbon fiber plates connected by aluminum parts. Likewise, the leg structures are made of carbon fiber plates connected by spacer bolts in the case of the upper leg and by a custom-designed plastic spacer in the case of the lower leg. Lastly, the feet of the robot are 3D-printed and exchangeable. This allows different material hardnesses to be tested for different substrates.

4.2. Actuator model and power consumption validation

The trajectory optimization formulation introduced in Section 3 is used to produce an energy-optimal bounding motion (for more details on the task, see Section 5.2). By tracking the optimal reference trajectory with the prototype, the gap between the model and reality is assessed, and the models are validated. Figure 4 shows that the actuator model with the identified parameters, closely predicts the total joint

[1] Video at https://peertube.laas.fr/w/jAfmfQD1mfeqhzxYHn1Uj3

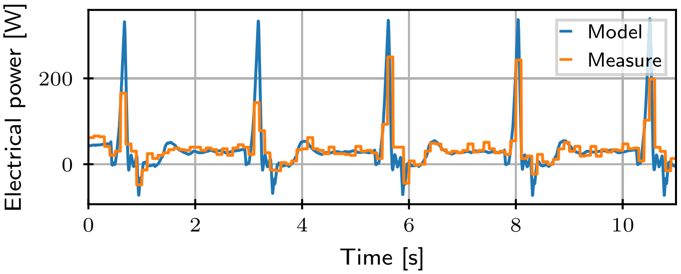


Figure 5. *The electrical power estimation P_e (blue) closely follows measured one $\hat{P}_e$ (orange).*

torque $\boldsymbol{\tau}$ (including joint friction $\boldsymbol{\tau}_f$) as in (17). A jumping trajectory cycle lasts 0.8 s, so it is repeated multiple times, with a phase in which the system resets to the initial position. To stabilize the trajectory, a PD joint position controller is used, with additional feedforward torques from the OCP. The value of the joint torques predicted by the model closely follows the measures, with the main difference in the flying phase [0.3 s, 0.5 s], which can be attributed to the unmodeled controller dynamics. In Fig. 5, the power prediction from joint data measurements (torque and velocity) is shown together with the measured data. The estimation of the total electrical power is given by $P_e = P_m + P_t + P_f$, with the notation introduced in Section 3.3. To compute the values, τ is inferred by our joint friction model. Figure 5 shows that the prediction, which solely uses joint measurements (velocities and commanded torques), follows the measurements of the electrical power provided by the power source, which is measured as the time average product of voltage V and current i, $\hat{P}_e = iV$ (at a lower sampling rate). Nonetheless, the integrated values of the total electrical consumption are accurate, despite the controller dynamics and the sim-to-real gap. These findings are reported for the energy-optimal trajectory in Table III. To give some context to these values, a heuristic baseline is used as a reference. Such baseline is based on a simple motion generation technique that is inspired by the energy-shaping approach [61], as implemented in Section II B of ref. [62]. The trajectory was tuned to produce a similar jump on the prototype (in terms of time horizon and base displacement). Simple parabolic trajectories based on accelerations are computed for each leg and the configuration in stance and flight phases are pre-defined. However, it is indeed a hand-tuned technique that is error-prone, time-consuming, and, most of all, sub-optimal (for energy efficiency). Trajectory optimization overcomes these weaknesses, and we compare the margin of improvement over the baseline method available. On this handcrafted trajectory, the knowledge of the actuator was applied to assess the electrical power consumption. It was found that the power consumption of the motion obtained with the heuristic was higher (30% increase) than the energy-optimal trajectories. Even if this result is not conclusive to claim the optimality (we just test the optimization output against a simple baseline), it is still an important step to confirm that our method can accurately estimate and minimize energy expenditure before including it in the co-design optimization.

5. Co-Design Optimization Results

5.1. Problem requirements and assumptions

High-level requirements for the platform are (i) to produce stable locomotion in the forward direction x, (ii) to be capable of dynamic motions along the z axis as shown in Fig. 6. In order to consider representative legged robot movements, we focus on the generation of (iii) stable and periodic motion patterns. Such movements need to be performed while being (iv) energy-efficient. Taking this into account, periodic bounding and backflip were selected as benchmark tasks to achieve.

Robot model. Figure 6 shows a sketch of the joint placements on a complete robot. The general design choice is to place the motors as close as possible to the base to limit the reflected inertia of the leg links.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Table III. *Energy consumption values for the jump.*

Case	Measured	Reference
Energy-optimal	39.69J	39.64J
Heuristic	56.75J	68.47J

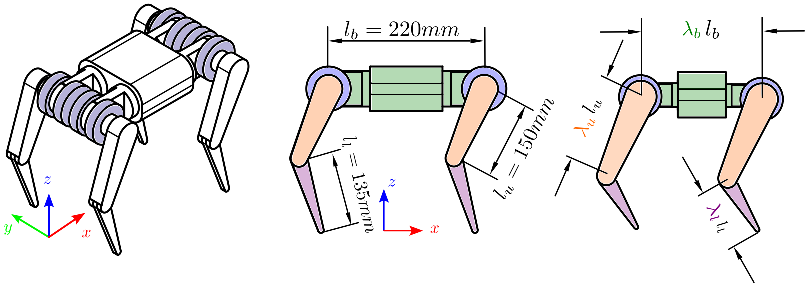


Figure 6. *Complete robot model (left), its planar simplification (center), and scaling of the base, upper leg, and lower leg links.*

Another preliminary design choice is to drive both the abduction joint and the hip joint directly, while using a belt transmission with a reduction factor of 2 for the knee joint. Since the motion of leg abduction in the lateral plane (y, z) is not needed for bounding or jumping, a planar model was used instead of the complete one. The motors are located on the robot's trunk. The abduction of the leg (rotation around x of the first leg joint) is blocked (the corresponding motor is located inside the base), while the motors actuating the hip and knee joints are shown in gray in Fig. 6. The robot model for this task exploits the symmetry of the motion with respect to the (x, z) plane. The dynamical equivalence between the 3D model and the planar one is ensured by lumping each link inertia and control effort on a unique joint for each symmetrical hip and knee. The command torque on the joints, and their limits, are then doubled and need to be equally divided into two legs when controlling the real system.

Structural scaling of the model.

The legs and torso are modeled with some fixed payloads, that correspond mainly to the mass of the motors (there is also a smaller contribution to the mass from embedded electronics). For the rest of the structure (fixating frames and FRP panels), we consider the effect of scaling and knowing that the rigidity of the system with respect to bending is much higher than with additive manufacturing, we can envision scaling up the link along its main nominal dimension with a factor λ (see Fig. 6). For the planar quadruped model of Fig. 6, three scaling factors are considered: λ_u, λ_l, and λ_b, respectively, for the upper leg, lower leg, and base of the robot. This scaling is just acting on the links' structure. The mass and dimension of the fixed payload (e.g., motors) do not scale with the rest of the rigid bodies (only the contribution depending on the geometry change does, for example, the attach point of the motor). The material density is assumed constant, and the section of the links is not modified. This scaling affects the link inertial parameters as follows:

- The mass scales linearly $\propto \lambda$.
- The center of mass position scales linearly $\propto \lambda$.
- Inertia: for the inertial parameters, each link geometry is simplified with box primitives, and each component of the inertia tensor is modified independently after the scaling. However, it is possible to intuitively envision the major contribution to the tensor. For this scaling, the effect on the inertia tensor is twofold: there is a purely geometric scaling with respect to the main link

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Table IV. *Properties of the motor selection integer variables.*

Motor type	**AK80-9**	**AK80-6**
Mass [kg]	0.48	0.48
Rotor inertia [kg m^2]	6.1E-05	6.1E-05
K [Nm/$\sqrt{W}$]	0.22	0.22
Gear reduction [-]	9	6
Nominal torque [Nm]	9.83	6.55
Peak speed [rad/s]	25.66	38.22

dimension ($\propto \lambda^2$), and a second one just related to the mass scaling ($\propto \lambda$). The overall scaling of the dominant inertia component is instead $\propto \lambda^3$. For the concentrated masses the additive contribution to the link inertia scales as λ^2

Design variables.

For both co-design tasks, we optimize over the same set of variables, which is here reported.

Continuous design variables. Starting from the nominal design, the following continuous design parameters are as follows:

- Lower leg link scaling $\lambda_l \in [0.5, 1.5]$
- Upper leg link scaling $\lambda_u \in [0.5, 1.5]$
- Base scaling $\lambda_b \in [0.5, 1.5]$

Discrete design variables. For the implementation, two promising motor candidates were selected from the off-shelf Antigravity AK series as reported in Table IV. In particular, among AK80-6 and AK80-9, these two motors differ mainly from the reduction of the integrated rotary gear, which is respectively 6 and 9. Negligible differences are found for the other parameters, especially concerning the motor constant and the winding resistance. In the co-optimization problem, the same leg design, and consequently actuator choice, is used for all four legs. The motor combinations for the hip and knee motors (respectively m_{hip}, m_{knee}) are then four. CMA-ES, which by default works on a continuous space, is modified to work on this discrete domain. This is not restrictive, as the discrete variables could also be handled by another GA selection, with minimal modifications to the framework. In our case, as the combinatorics of the problem are rather small, the discrete variables are internally considered continuous, and a remapping strategy is employed to pass from the continuous to the discrete counterpart before solving the problem. Thanks to this remapping, all of the motor characteristics are found with the catalog value without the need for an explicit parametrization as in refs. [22, 23, 33, 34]. This mechanism enables optimization even in cases in which a continuous parametrization is not viable (e.g., motor technologies are rather different from each other).

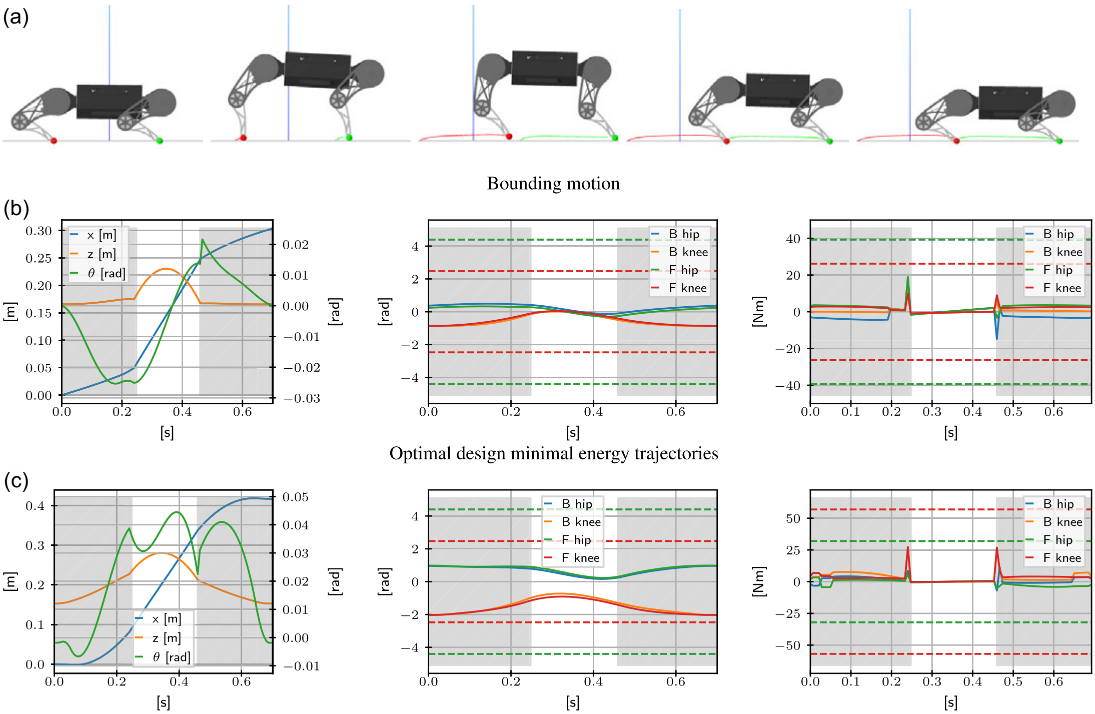
Actuator choice.

The actuator properties are taken into account by modifying the robot dynamics, the constraints of the OCP, and the cost function. The main effects of the actuator are as follows:

Joint limits. The choice of a motor and its transmission directly impacts the effort limit (torque) and speed limit which is achievable by the controlled joint.

Inertia. The added rotor inertia is considered in the model via the technique explained in Ch. 9.6 of ref. [56]: the rotor inertia is multiplied by the value of the squared reduction and added to the corresponding diagonal element of the joint space mass matrix.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press



Bounding motion

Optimal design minimal energy trajectories

Nominal design minimal energy trajectories

Figure 7. *Bounding task: (a) shows the different motion phases. Trajectories for the optimal and the nominal designs are respectively shown in (b) and (c). In both, from left to right, the plots show base, joint positions, and joint torque trajectories. Contact phases are highlighted with gray background.*

Transmission friction. Given a motor and its transmission, the overall viscous and Coulomb friction are considered in the cost function that minimizes the overall energy, following the same approach as in ref. [33].

Motor placement. The motor masses are also taken into account by the structural base scaling. Each motor is modeled as a localized mass, and its contribution adds to the parent link mass and rotational inertia.

Actuator bandwidth. From the current technology of the actuator and testing, the cutoff frequency can be estimated as $f_c \approx 20Hz$. This is valid for both motors, and such choice constrains the torque trajectory conservatively.

5.2. Co-design for bounding

The first optimized task is a bounding motion, where the robot must perform a jump of at least 0.30 m. The cyclic constraints enforce the robot state to be equal at the beginning and the end of the trajectory, except for the base x position. Finally, a constraint is added to obtain zero joint velocities at the start and end of the trajectory. In this way, the system consumes just the energy required to perform the jump and decelerate to a full stop in the final part of the trajectory. The phases of such movement are as follows (Fig. 7(c)):

- *Double support*, with all feet in contact with the ground.
- *Flying phase*, with no contact with the ground.
- *Double support*, with all feet again in contact with the ground.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

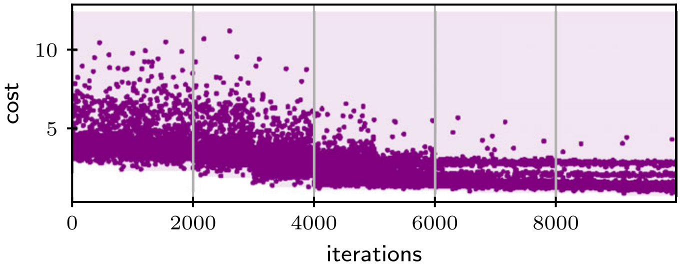


Figure 8. *Convergence of the algorithm along the evolution of the populations.*

This task is symmetrical, meaning that the time left for each contact phase is the same. For the overall problem, the time window for each cycle of the jump is 0.7s, and the total number of nodes for the optimization horizon is 100.

Outer loop hyper-parameters. For this optimization, the CMA-ES algorithm is initialized to evolve for 10 times a population of 1000 different individuals (different combinations of the design parameters). Depending on its complexity, each OCP problem's computation time varies between ≈ 10 s and 10 min. Thanks to parallelization, the whole optimization routine is carried out in ≈ 10 h. This time is comparable to the clock time of our previous contributions [33, 34], but for a much more complex system and task (more DoFs) and more accurate model of the system and dynamics (which is being used in the OCP). Figure 8 shows that this is sufficient to reach stationary values in the cost. It is clear from the trends that there is a diminishing return in exploring further combinations of parameters. In particular, in the same figure, we see different bands, which correspond to the various optimal designs for the 4 combinations of the motors. For simplicity, in the following the combinations will be called AK80 9-9, AK80 9-6, AK80 6-9, and AK80 6-6, where the first index is for the hip joint and the second for the knee joint.

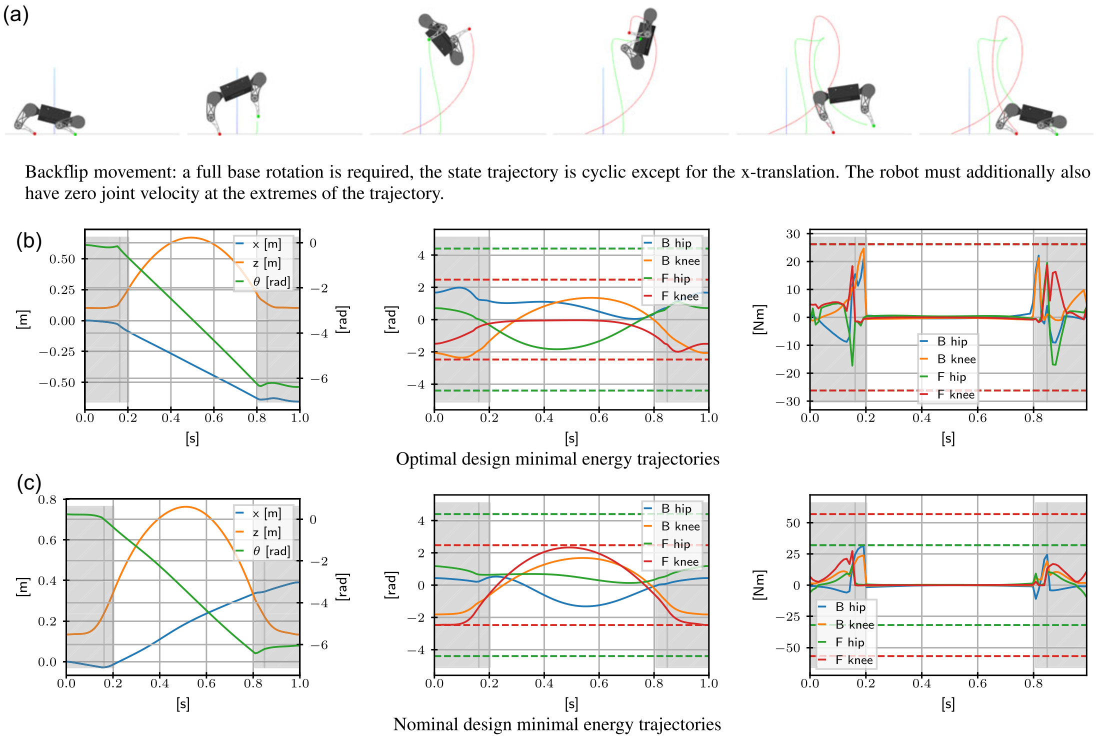
Discussion. For this task, the optimal design is obtained for the values reported in Table V. We see that, with respect to the nominal leg design, the best solution is found for a smaller robot.

According to Table V, the optimization selects as best suited a smaller robot, with a different scaling of the thigh λ_u and shank λ_l, in particular $\lambda_u/\lambda_l = 1.46$. For jumping forward, it seems then that robots with longer thighs are performing better. The optimal solution is chosen to be very close to the lower bound of the variables λ_b, λ_l. An additional effect of the choice of the base can be observed in Fig. 7(b): when the base scaling is reduced (Table V), the trajectories of the knee and hip joints are showing a higher degree of similarity. In the nominal case, the joint position is reaching the position limits of the actuator, which is no longer the case with the optimized hardware. Basically, we can explain this result as follows. The optimal quadruped for bounding tends to be shaped as a planar biped: since there is no advantage in carrying additional mass from an energetic point of view, the base length is chosen as short as possible. From the joint positions of the nominal design (Fig. 7(c)), the knee stopper can partially limit the robot's motion. So, finding a solution that does not impose a limitation would be advisable. For both designs, the optimal joint trajectories are smooth and not hitting the velocity bounds. So, concerning the actuator selection for this task, a motor capable of quick motions is not really necessary. Conversely, the choice of a higher gear ratio allows to exert larger torques and to greatly decrease the Joule consumption. To produce the same output torque (as the motor constant is the same) the ratio of the Joule dissipation of the motor types AK80-9 and AK80-6 are equal to the quotient of the square of their gear ratio, so 2.25, even if a higher reduction also impacts the reflected inertia, reducing the actuator transparency and joint maximum speed. For bounding the higher gear ratio is selected.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Table V. *Results of the optimization for the bounding task.*

	Nominal	Optimized
Cost $\mathcal{L}$	1.78	0.41
λ_u	1.0	0.752
λ_l	1.0	0.514
λ_b	1.0	0.512
m_{hip}	AK80-6	AK80-9
m_{knee}	AK80-6	AK80-9



Backflip movement: a full base rotation is required, the state trajectory is cyclic except for the x-translation. The robot must additionally also have zero joint velocity at the extremes of the trajectory.

Optimal design minimal energy trajectories

Nominal design minimal energy trajectories

Figure 9. *Backflip task: (a) shows the different motion phases. Trajectories for the optimal and the nominal designs are respectively shown in (b) and (b). In both, from left to right, the plots show base, joint positions, and joint torque trajectories. Contact phases are highlighted with a gray background.*

5.3. Co-design for backflip

As a second task, we present the result of a backflip optimization, as shown in Fig. 9. This motion was selected as a complex and dynamic task, exploiting the dynamics of the whole system. The robot starts with zero velocity and has to perform a full rotation of the base before landing. In the landing phase, the excess velocity needs to be damped to reach a full stop at the end of the trajectory. For this task, all joint positions except the base x component need to be equal at the beginning and the end of the trajectory. For this motion, the total time to perform the task is 1 s. As represented in Fig. 9(a), the different phases are as follows.

- *Double support*, with all the feet in contact with the ground. In this phase, the motors need to accelerate the base to produce enough vertical velocity to break the contact with the ground.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

Table VI. *Results of the optimization for the backflip.*

	Nominal	**Optimized**
Cost $\mathcal{L}$	5.21	0.67
λ_u	1.0	0.50
λ_l	1.0	0.689
λ_b	1.0	1.07
m_{hip}	AK80-6	AK80-6
m_{knee}	AK80-6	AK80-6

Moreover, the applied forces need to generate enough momentum for the upcoming rotation of the base.

- *Single support*, with the front legs taking off. This phase is added to allow the robot to start the rotation of the base and still push the ground with the back legs.
- *Flying phase*, in which there is no foot in contact with the ground and the base is following a ballistic movement. The motion of the legs is not contributing to the jump but is useful to get the feet in the right position before landing (preparing for the impact phase).
- *Single support*, with the front legs touching the ground first.
- *Double support*, with the rear legs reestablishing contact with the ground. The contact needs to be stable, so the forces are inside the friction cone and the motors bring the robot to a full stop at the end of the trajectory.

Outer loop hyper-parameters. CMA-ES is initialized so that each generation is made up of 10^3 individuals, and the number of overall evolutions of the population is fixed to 10. In this problem, as the task is more challenging, some designs could not physically satisfy the constraints and perform the motion within the problem constraints. *IpOpt* provides debug information on the infeasibility of the problem. If an individual is unfeasible, an arbitrary high-cost value, higher than the other feasible designs, is assigned to it. The outer loop algorithm is elitist, meaning that when generating a new population it will automatically discard the outlier designs.

Discussion. The results of this optimization are reported in Table VI. Running the optimization routine, we notice that the leg size is reduced while the base dimensions are slightly increased.

The optimization selects a smaller robot, but interestingly a different optimal scaling of thigh λ_u and shank λ_u is found (with ratio $\lambda_u/\lambda_l = 0.73$) with respect to the bounding task. For backflips, it seems then that robots with longer shanks are performing better. The optimal solution is very close to the lower bound of the variable λ_u. Figure 9(b) and (c) report the optimal and nominal design trajectories. The optimal base scaling is obtained with a bigger base without reaching the upper bound. This can be explained as there is a tradeoff between the base inertia and the capability to apply momentum to perform a full base rotation around θ of -2π rad. For the same applied contact forces, the longer the base, the easier the backflip can be performed. However, there is still a tradeoff as a bigger base increases also the inertia of the rigid body. For the backflip, it seems that the most critical constraint is the maximum joint velocity. The nominal design, featuring a higher reduction, can exert more torque but reaches the joint velocity limit. A motor that can produce faster motion is needed for task completion. The optimization leads to a smaller reduction to achieve a higher joint velocity, differently from what was obtained for the bounding task. In this case, as the motion is quicker, the effect of the rotor inertia is higher and a smaller reduction helps in accelerating the joint. Compared to the torque required for bounding, in this case, saturation is reached. In Fig. 9(b) and (c), the low-pass filter effect can be noticed from the smooth torque trajectories that do not exceed the upper and lower torque bounds of the actuator (shown with dotted lines). For this task, the knee joint is reaching the position limits of the actuator, in both designs (nominal and optimized). Such limit needs to be taken carefully considered for the final robot design.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

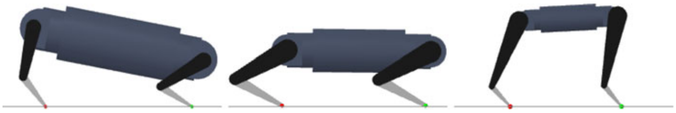


Figure 10. Center: nominal, left: optimized backflip, right: optimized bounding.

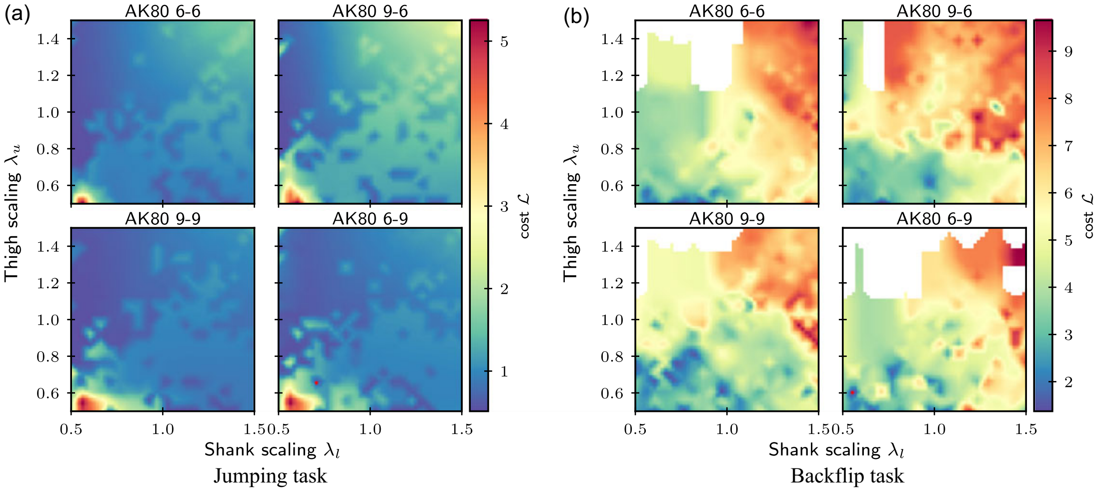


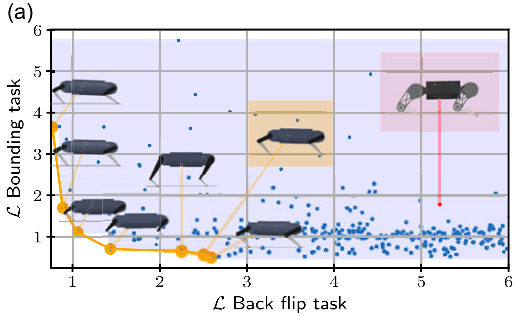
Figure 11. Cost landscape for the different motor combinations and link scaling. In (b), white regions are associated with unfeasible problems, which occur consistently when the robots have short shanks and long thighs.

5.4. Landscape analysis for multiple objectives

As rather different designs were produced for the two tasks by the co-design optimization (Tables V and VI, Fig. 10), an additional grid search was performed to better understand the impact of the design selection. In this case, the base was kept to the nominal length $\lambda_b = 1$, and leg design alone was studied for the two tasks presented before. The scaling of the upper and lower leg link is then modified together with the motor selection. For the scaling, a uniform grid of 50×50 was studied within the range [0.5, 1.5]. The results are depicted in Fig. 11(a) and (b) for the bounding and the backflip tasks, respectively. The plots show the value of the cost against the scaling of the upper link λ_u and lower link λ_l once a specific motor combination is chosen. From the trends of the optimal value $\mathcal{L}$, some orthogonality emerges between the two tasks in the design space.

With the values obtained from the grid search, the Pareto front is approximated (Fig. 12(a)) for the two different task costs. The result is reported in Fig. 12(b): it constitutes a reduced set of candidates that can perform both bounding and backflip reasonably well. As a second-order criterion, designs that involve fewer modifications to the nominal prototype are preferred. Practical considerations drive this: modifying the shank link is easier than the thigh, as the modification of the latter involves a re-design of the transmission, which is more challenging. Moreover, an optimized robot for bounding is preferred if this implies a sacrifice of performance for backflips (locomotion on the x direction is the main movement mode). So the closest options are the 2nd and 3rd rows of Table (b) in Fig. 12, which use the same motors and a lower link very close to the nominal one. Among the two, the one with the scaling parameter of the thigh at $\lambda_u = 0.66$ was selected. The chosen design decreases the cost for the motion, as shown in Fig. 12(a). The relative improvement of this design with respect to the nominal one is 52% for backflip and 67% for bounding. Some performance was sacrificed for the sake of versatility as, for a single task, we found improvements of 87% and 77% respectively.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press



Different design on the Pareto front

(b)

λ_u	λ_l	m_{hip}	m_{knee}	$\mathcal{L}$ Bck	$\mathcal{L}$ Bnd
0.50	0.87	AK80-6	AK80-6	1.06	1.11
0.50	1.03	AK80-6	AK80-6	2.59	0.50
0.66	1.03	AK80-6	AK80-6	2.50	0.57
0.71	0.55	AK80-6	AK80-6	0.77	3.65
0.50	0.71	AK80-9	AK80-9	0.89	1.70
0.61	0.97	AK80-9	AK80-9	1.43	0.70
1.08	0.50	AK80-9	AK80-9	2.26	0.65

Pareto candidates values

Figure 12. *Pareto front approximation for the two tasks' cost. The designs are visually superimposed in (a). The one highlighted in orange is the design which requires the least modifications to the nominal prototype, shown in red. In the side Table (b) the different points are reported.*

6. Conclusions and Future Work

In this work, our co-design framework was improved to gain completeness and versatility, and it was applied to the optimization of a quadruped robot developed at DFKI-RIC. The development of a more robust and parallelizable bi-level co-design scheme was achieved. A more versatile OCP formulation with equality and inequality constraints was used to minimize energy consumption, imposing actuation bandwidth and motion cyclicity constraints. The trajectory optimization output was validated on the current prototype robot for bounding, showing lower energy consumption and accurate friction and power estimation. The optimization strategy was also modified to include discrete variables and it was used to optimize bounding and backflip tasks. These two cyclic tasks were selected to represent different key motion capabilities, even if more complex motions could also be considered. In an initial phase, hardware was optimized for each of the two tasks separately. Since rather different designs were obtained, a Pareto set was approximated to select a versatile and efficient tradeoff. The main limitation of the method remains the computational cost of the OCP resolution, which is a general bottleneck for co-design. In future work, we plan to modify the hardware according to our findings. This last step will close the loop between optimization and fabrication. It is a crucial challenge, because, even though key information can be obtained from such a computational framework, expert knowledge is needed both to select the final design and to physically build and test it.

Author contributions. G. Fadini, S. Kumar, P. Soueres, T. Flayols, A. Del Prete conceived and designed the study. G. Fadini, implemented the bi-level optimization framework, performed the data analysis, and wrote the article. J. Carpentier implemented advanced features in `pinocchio` fundamental to implement the "OCP." R. Kumar performed the experimental testing and data collection. All the authors took part in proofreading.

Financial support. This work was supported by: the French government as part of the ROBOTEX 2.0 program (ANR-10-EQPX-44-01, TIRREX-ANR-21-ESRE-0015) and the "Investissements d'avenir" program, ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and by the EU through the AGIMUS project (GA no.101070165). The 2nd and 3rd authors acknowledge the support of M-RoCK (FKZ 01IW21002) project funded by the German Aerospace Center with federal funds from the Federal Ministry of Education and Research.

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

Acknowledgements. The authors thank Heiner Peters and the DFKI Underactuated Robotics Lab, for the development and system identification of the prototype and Guilhem Saurel and the IDEA team of LAAS-CNRS, for the guidance using the HPC cluster.

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

References

[1] P. Wensing, "The rapid rise of quadruped robots [young professionals]," *IEEE Robot. Autom. Mag.* **28**(3), 168–168 (2021).

[2] Anybotics. Anymal X. *Available at:* https://www.anybotics.com/anymal-x-the-worlds-first-ex-proof-legged-robot/.

[3] B. Dynamics, Spot specifications. *Available at:* https://support.bostondynamics.com/s/article/Robot-specifications.

[4] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer and M. Hoepflinger, "ANYmal - a Highly Mobile and Dynamic Quadrupedal Robot," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (2016).*

[5] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing and S. Kim, "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (2018).*

[6] D. J. Hyun, S. Seok, J. Lee and S. Kim, "High Speed Trot-Running: Implementation of a Hierarchical Controller Using Proprioceptive Impedance Control On the MIT Cheetah," *Int. J. Robot. Res*. vol. 33. Sage Publications: UK, London, England (2014).

[7] H.-W. Park, P. M. Wensing and S. Kim, "High-speed bounding with the MIT cheetah 2: Control design and experiments," *The International journal of Robotics Research 2017* **36**(2), 167–192 (2017).

[8] Unitree, Unitree A1. *Available at:* https://unitreerobotics.net/robotdog/unitree-a1.

[9] B. Katz, J. D. Carlo and S. Kim, "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control," **In:** *IEEE International Conference on Robotics and Automation (2019).*

[10] J.-H. Park and H. Asada, "Concurrent design optimization of mechanical structure and control for high speed robots," *American Control Conference*, San Francisco, CA, USA (1993), pp. 2673–2679.

[11] J. A. Reyer and P. Y. Papalambros, "Combined optimal design and control with application to an electric DC motor," *J. Mech. Des*. **124**(2), 183–191 (2002).

[12] S. Rezazadeh, A. Abate, R. L. Hatton and J. W. Hurst, "Robot leg design: A constructive framework," *IEEE Access* **6** (2018).

[13] S. Rezazadeh and J. W. Hurst, "On the Optimal Selection of Motors and Transmissions for Electromechanical and Robotic Systems," **In:** *IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (2014).*

[14] C. Semasinghe, D. Taylor and S. Rezazadeh, "A Unified Optimization Framework and New Set of Performance Metrics for Robot Leg Design," **In:** *IEEE Int. Conf. On Robotics and Automation*

[15] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang and S. Kim, "Proprioceptive actuator design in the MIT cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Trans. Robot*. **33**, 509–522 (2017).

[16] K. D. Mombaur, R. W. Longman, H. G. Bock and J. P. Schlöder, "Open-Loop Stable Running," **In:** *Robotica 2005*. vol. 23 (Cambridge University Press, 2005).

[17] G. Buondonno, J. Carpentier, G. Saurel, N. Mansard, A. De Luca and J.-P. Laumond, "Actuator Design of Compliant Walkers Via Optimal Control," **In:** *IEEE/RSJ International Conference on Intelligent Robots & Systems (2017).*

[18] G. Saurel, J. Carpentier, N. Mansard and J.-P. Laumond, "A Simulation Framework for Simultaneous Design and Control of Passivity Based Walkers," **In:** *IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (2016).*

[19] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik and B. Bickel, "Computational Design of Mechanical Characters," *ACM Trans. Graph*., vol. 32(4).

[20] K. M. Digumarti, C. Gehring, S. Coros, J. Hwangbo and R. Y. Siegwart, "Concurrent Optimization of Mechanical Design and Locomotion Control of a Legged Robot," **In:** *Mobile Service Robotics* (2014).

[21] A. Spielberg, B. Araki, C. Sung, R. Tedrake and D. Rus, "Functional co-optimization of articulated robots," *arXiv:* 1707.06617 *[cs]* (2017).

[22] Y. Yesilevskiy, Z. Gan and C. D. Remy, "Energy-optimal hopping in parallel and series elastic one-dimensional monopeds," *J. Mech. Robot*. **10**(3), 031008 (2018).

[23] Y. Yesilevskiy, Z. Gan and C. Remy, "Optimal Configuration of Series and Parallel Elasticity in a 2D Monoped," **In:** *IEEE International Conference on Robotics & Automation (2016).*

[24] S. Ha, S. Coros, A. Alspach, J. Kim and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *Int. J. Robot. Res*. **37**(13-14), 1521–1536 (2018).

[25] M. Chadwick, H. Kolvenbach, F. Dubois, H. F. Lau and M. Hutter, "Vitruvio: An Open-Source Leg Design Optimization Toolbox for Walking Robots," **In:** *IEEE Robotics and Automation Letters* (2020).

[26] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau and S. Vijayakumar, "A Versatile Co-Design Approach for Dynamic Legged Robots," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (2022).*

[27] G. Bravo-Palacios, A. D. Prete and P. M. Wensing, "One Robot for Many Tasks: Versatile Co-Design through Stochastic Programming," **In:** *IEEE RAL 2020*. vol. 5 (2020) pp. 1680–1687.

[28] G. Bravo-Palacios, G. Grandesso, A. D. Prete and P. M. Wensing, "Robust co-design: Coupling morphology and feedback design through stochastic programming," *J. Dynam. Syst Meas. Contr.* **144**(2) (2022). https://asmedigitalcollection.asme.org/dynamicsystems/article/144/2/021007/1119647/Robust-Co-Design-Coupling-Morphology-and-Feedback

[29] G. Bravo-Palacios and P. M. Wensing, "Large-Scale ADMM-Based Co-Design of Legged Robots," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (2022).*

[30] W. Bosworth, S. Kim and N. Hogan, "The MIT Super Mini Cheetah: A Small, Low-Cost Quadrupedal Robot for Dynamic Locomotion," **In:** *IEEE SSRR* (2015).

[31] F. Girlanda, L. Maywald, S. Kumar and F. Kirchner, "Robust Co-Design of Canonical Underactuated Systems for Increased Certifiable Stability," **In:** *IEEE ICRA (2024).*

[32] L. J. Maywald, F. Wiebe, S. Kumar, M. Javadi and F. Kirchner, "Co-Optimization of Acrobot Design and Controller for Increased Certifiable Stability," **In:** *IEEE International Conference on Intelligent Robots and Systems (2022).*

https://doi.org/10.1017/S0263574724000730 Published online by Cambridge University Press

[33] G. Fadini, T. Flayols, A. D. Prete, N. Mansard and P. Souères, “Computational Design of Energy-Efficient Legged Robots: Optimizing for Size and Actuators,” **In:** *IEEE International. Conference on Robotics & Automation (2021).*

[34] G. Fadini, T. Flayols, A. D. Prete and P. Souères, “Simulation Aided Co-Design for Robust Robot Optimization,” **In:** *IEEE Robotics and Automation Letters*. vol. 7 (IEEE, 2022).

[35] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri and A. J. Ijspeert, “Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot,” *Int. J. Robot. Res*. **32**(8), 932–950 (2013).

[36] N. Hansen, The CMA evolution strategy: A tutorial arXiv:1604.00772.

[37] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program* **106**, 25–57 (2006).

[38] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, “CasADi A Software Framework for Nonlinear Optimization and Optimal Control,” **In:** *Mathematical Programming Computation (2019)*. vol. 11-(1). Springer.

[39] J. Carpentier and N. Mansard, “Analytical Derivatives of Rigid Body Dynamics Algorithms,” **In:** *Robotics: Science and Systems (2018).*

[40] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse and N. Mansard. “The Pinocchio C++ Library – A Fast and Flexible Implementation of Rigid Body Dynamics Algorithms and Their Analytical Derivatives,” **In:** *IEEE/SICE International Symposium on System Integration (2019).*

[41] D. Mayne, “A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems,” *Int. J. Control* **3**(1), 85–95 (1966).

[42] Y. Wang and S. Boyd. “Fast Model Predictive Control Using Online Optimization.” **In:** *IFAC Proceedings Volumes 2008*, vol. 41. IFAC World Congress.

[43] E. A. Yildirim and S. J. Wright, “Warm-start strategies in interior-point methods for linear programming,” *SIAM J. Optim.* **12**, 782–810 (2002).

[44] K. Mombaur, H. Bock and R. Longman, “Stable, Unstable and Chaotic Motions of Bipedal Walking Robots Without Feedback,” **In:** *2nd Int. Conf. Control of Oscillations and Chaos*. vol. 2 (2000).

[45] K. H. Koch, K. Mombaur and P. Souères. “Optimization-Based Walking Generation for Humanoid Robot,” **In:** IFAC Proceedings Volumes 2012, 10th IFAC Symposium on Robot Control, vol. 45 (2012).

[46] C. Chevallereau and Y. Aoustin, “Optimal Reference Trajectories for Walking and Running of a Biped Robot,” **In:** *Robotica 2001*. vol. 5 (Cambridge University Press, 2001).

[47] I. Mordatch, E. Todorov and Z. Popovic, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Trans. Graph.* vol. 31 (2012)

[48] M. Posa, S. Kuindersma and R. Tedrake, “Optimization and Stabilization of Trajectories for Constrained Dynamical Systems,” **In:** *IEEE International Conference on Robotics & Automation (2016).*

[49] M. Posa and R. Tedrake, “Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact,” **In:** *Algorithmic Foundations of Robotics X* (Springer, 2013).

[50] E. Todorov, “A Convex, Smooth and Invertible Contact Model for Trajectory Optimization,” **In:** *IEEE International Conference on Robotics & Automation (2011).*

[51] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, “Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point,” **In:** *IEEE International Conference on Robotics & Automation*, vol. 2 (2003).

[52] S. Kajita, T. Nagasaki, K. Kaneko and H. Hirukawa, “ZMP-based biped running control,” *IEEE Robot. Autom. Mag.* **14**(2), 63–72 (2007).

[53] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi, “Biped Walking Stabilization Based On Linear Inverted Pendulum Tracking,” **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010).

[54] H. Dai, A. Valenzuela and R. Tedrake. "Whole-Body Motion Planning with Centroidal Dynamics and Full Kinematics." **In:** *IEEE-RAS International Conference on Humanoid Robots* (2014).

[55] D. E. Orin, A. Goswami and S.-H. Lee, “Centroidal Dynamics of a Humanoid Robot,” **In:** *Autonomous Robots*. vol. 35.

[56] R. Featherstone. *Rigid Body Dynamics Algorithms* (Springer New York, NY, 2016).

[57] C. Hubicki, M. Jones, M. Daley and J. Hurst, “Do Limit Cycles Matter in the Long Run? Stable Orbits and Sliding-Mass Dynamics Emerge in Task-Optimal Locomotion,” **In:** *IEEE International Conference on Robotics and Automation (2016).*

[58] R. Grandia, F. Farshidian, R. Ranftl and M. Hutter, “Feedback MPC for Torque-Controlled Legged Robots,” **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (2019).*

[59] N. K. Gupta, “Frequency-shaped cost functionals - extension of linear-quadratic-gaussian design methods,” *J. Guid. Cont. Dynam.* **3** (1980).

[60] J. Pieper, mjbots quad. *Available at:* https://github.com/mjbots/quad.

[61] R. Tedrake, *Underactuated robotics - algorithms for walking, running, swimming, flying, and manipulation* (2023).

[62] R. Soni, D. Harnack, H. Isermann, S. Fushimi, S. Kumar and F. Kirchner. “End-to-End Reinforcement Learning for Torque Based Variable Height Hopping,” **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(2023) pp. 7531–7538.

Cite this article: G. Fadini, S. Kumar, R. Kumar, T. Flayols, A. Del Prete, J. Carpentier and P. Souères (2024). “Co-designing versatile quadruped robots for dynamic and energy-efficient motions”, Robotica **42**, 2004–2025. https://doi.org/10.1017/S0263574724000730