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

Gabriele Fadini, Shivesh Kumar, Rohit Kumar, Thomas Flayols, Andrea del Prete, Justin Carpentier, Philippe Souères

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

G. Fadini[1,⋆], S. Kumar[2], R. Kumar[2], T. Flayols[1], A. Del Prete[3], J. Carpentier[4], P. Souères[1]

***Abstract*—This paper presents a bi-level optimization framework to concurrently optimize a quadruped hardware and control policies for achieving dynamic cyclic behaviors. The long-term vision to drive the design of dynamic and efficient robots by means of computational techniques is applied to improve the development of a new quadruped prototype. The scale of the robot and its actuators are optimized for energy efficiency considering a complete model of the motor, that includes friction, torque, and bandwidth limitations. This model is used to optimize the power consumption during bounding and backflip tasks and is validated by tracking the output trajectories on the first prototype iteration. The co-design results show an improvement of up to 87% for a single task optimization. It appears that, for jumping forward, robots with longer thighs perform better, while for backflips, longer shanks are better suited. To understand the trade-off between these different choices, a Pareto set is constructed to guide the design of the next prototype.**

***Index Terms*—Methods and Tools for Robot System Design, Optimization and Optimal Control, Simulation and Animation, Mechanism Design**

I. 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.

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 (DLR) with federal funds from the Federal Ministry of Education and Research (BMBF).

⋆ Corresponding author gfadini@laas.fr
[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, Italy
[4] INRIA and Département d'informatique de l'ENS, Paris, France

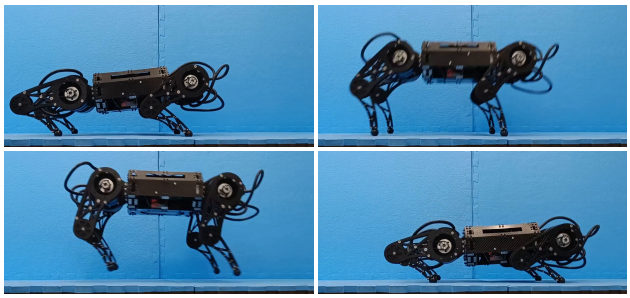


Fig. 1. **Quadruped prototype** bounding tests at DFKI-RIC [1].

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 [8]	Unitree A1 [9]
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 Stairs†	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.

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 Zurich, the Cheetah series [5]–[7] from MIT. Especially after the seminal work on the open-source MIT mini-Cheetah robot [8], which has demonstrated back flips, and other highly athletic behaviors, various other smaller-sized quadrupedal platforms became 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, the ability to perform a backflip). On the other hand, quadrupeds with larger body lengths can climb standard stairs, which is very useful in deploying them in real-world environments developed for human accessibility. A natural question arises: how can we design quadruped robots that can optimally perform a range of dynamic movements?

Several contributions already dealt with the problem of robot co-design. In [10], the motion of a biped was optimized together with its kinematic parameters to produce stable running by using local trajectory optimization coupled with a

[1] Companion video at https://peertube.laas.fr/w/iUscYk7iigi4v3sgk97XxV

genetic optimization for the hardware parameters. In [11], [12], passive walkers actuators were optimized for cyclic motions. Design kinematic parameters were chosen in order to produce smooth motion for mechanical avatars in [13], while in [14] the leg design of the StarlETH was selected to optimize peak speed. In [15] several simple legged robots were designed in a single NLP problem where the hardware was optimized at the same level of the motion. The method allowed to include hard constraints on task fulfillment. Other work focused on the optimality of motion and design, for instance in [16], [17] monopeds were designed to minimize different cost functions, targeting energy efficiency. In [18], 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 [19], 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 can not fully exploit the system dynamics. In [20] 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. In [21], 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 [22]. 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 [23], [24]. Some preliminary results in integrating the trajectory stabilization at the design level for simple underactuated systems can be found in [25]. In [26], [27] two successive works where conducted to develop a generic framework to cover legged robots co-design combining trajectory optimization and genetic algorithms. Several model-based paradigms for legged systems hardware selection, featuring several design criteria, have been proposed in [28]–[31]. Additional experimental work validating hardware selection choices was performed in [32]. 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 [26], [27] 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 robust optimal control problem resolution and parallelization of the framework to improve both accuracy and computation times required by the higher complexity of the platform and tasks to optimize.
- Complete modeling of actuator bandwidth handled as motion constraints.
- Consideration of optimal cyclic movement patterns (e.g. bounding and back-flip). In particular, to the best authors' knowledge, multiple dynamic and periodic movements have never been analyzed with co-design.
- Real hardware validation of energy and friction models of the actuator used in trajectory optimization.

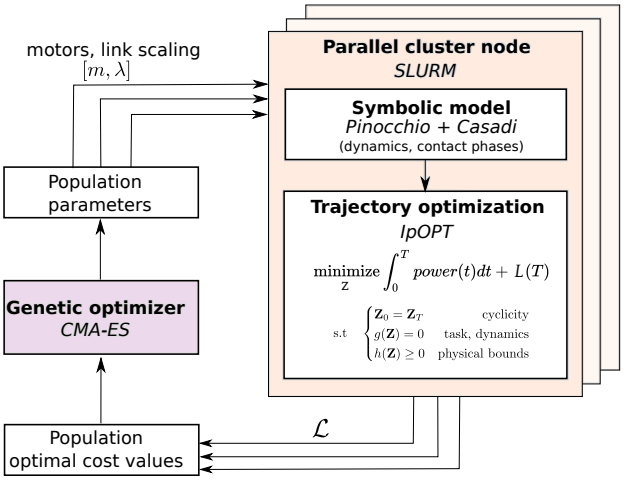


Fig. 2. **Overview of the approach.** Stack of the parallelized bi-level optimization scheme with arbitrary hard constraints on the primal optimization variables.

The paper is organized as follows. The rationale and theoretical aspects of the framework are outlined in Section II. In Section III we focus on the optimal control problem formulation. Here the actuator model is presented together with its impact on the constraints (bandwidth limitation) and cost function (electrical power consumption). In Section IV-A 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 V. In particular, in Section V-B and V-C 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 V-D.

II. METHODOLOGY

A. Co-design framework structure and characteristics

Fig. 2 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.

a) Outer loop (genetic algorithm): To optimize over the hardware parameter space, we use a gradient-free, population-based stochastic optimization CMA-ES [33]. 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 [18], which depend on an initial guess. The discrete motor selection is optimized together with the other design variables by CMA-ES, which by default works on a continuous space. All the design-related quantities are continuous, and so a remapping strategy is employed to pass from the continuous variables associated with the motor selection to their discrete counterpart before solving the problem. CMA-ES optimizes over a set of continuous variables exploring all the motor combinations, and internally, before computing the associated OCP the projection to the integer is performed. Thanks to this remapping, all of the motor characteristics are found with the catalog value without the need for an explicit parametrization as used in [16], [17], [26], [27]. This mechanism is necessary when the motor technologies are rather different from each other or when a parametrization is not viable. 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. 2.

b) 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* [34] for solving the OCP. In our previous work [26], [27], 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 to 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* [35] and *Pinocchio* [36], [37]. 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 [38], as the specific sparsity pattern of the OCP is not exploited. Depending on its complexity, each OCP problem's computation time varies between ≈ 10 s and 10 min. Moreover, the addition of inequality and equality constraints drastically increases the complexity. The time-sparsity pattern in the Hessian matrix is partially recovered by the linear solver `MA57` [39], which we selected because of its efficiency and robustness.
- warmstart capability; because of the barrier initialization, interior point methods are more difficult to warm start. This usually limits the re-using of previously computed solutions to solve a new problem instance [40], [41].

Table II compares our method with other state-of-the-art co-design strategies. The work that is more similar to ours in terms of optimized platform and trajectories is [21]. The main advantages of our approach are the bi-level structure of the optimization which allows discrete variables to be optimized, and the capability to handle actuator bandwidth limitations. Finally the use of CMA-ES for the outer-loop enables to explore globally the combination of hardware parameters, while in general gradient-based methods are more impacted by initialization. These features provide an advantage with respect to state-of-the-art methods for practical design problems.

TABLE II
COMPARISON BETWEEN VARIOUS STATE-OF-THE ART CO-DESIGN APPROACHES.

Contribution	[19]	[18]	[15]	[10]	[14]	[21]	[20]	[26]	[27]	Ours
Whole-body*				✓	✓	✓	(✓)	✓	✓	✓
Bandwidth										✓
Hard constraints				✓	✓	✓				✓
Cyclicity				✓		✓				✓
Scaling	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Energy optimization	✓					✓	✓	✓	✓	✓
Global exploration				✓				✓	✓	✓
Mixed-integer										✓
Robustness						✓			✓	
Controller						✓	✓		✓	
Hardware validation			✓		✓					✓

* Exact dynamics instead of kinematic/reduced models.

III. TRAJECTORY OPTIMIZATION PROBLEM

A. Trajectory optimization formulation

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 [42]–[45]. 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.

a) Variables: 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:

- $\mathbf{F}$: contact force vector, which stacks the contact forces $\mathbf{f} = \bigcup_C \bigcup_p (\mathbf{f}_{c,p} \in \mathbb{R}^{n_c})$, 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.

- $\mathbf{\Gamma}$: slack variable vector, which collects the contact slack variables $\boldsymbol{\gamma} = \bigcup_C \bigcup_p (\boldsymbol{\gamma}_p \in \mathbb{R}^{n_c})$, following the formulation in [45], to impose constraints on the feet position and velocity. These variables are introduced only in the contact phases to avoid contact drift.

b) Model choice: The considered augmented set of variables, at torque and acceleration level, is motivated by the intention of imposing physically driven constraints on the trajectory considering the physical limitations of the actuator. This is not directly possible in the case of simplified models such as Linear Inverted Pendulum [46], [47], Spring-Loaded Inverted Pendulum [48] or centroidal model [49], [50], which do not include the joint torques. Finally, highly-dynamic behaviours are difficult to discover as they are often far from the simplified model assumptions.

c) Contact phases: This study is limited to the case of trajectories with pre-specified phases and timing (the sequence of contacts is fixed a-priori). We follow a holistic approach inspired by [51] (and later [10], [52]) where the motion of a biped is synthesized by imposing periodic constraints on the trajectory. As in [53], the joint trajectories of a planar biped are optimized to obtain cyclic behaviors imposing contact constraints and joint limits. 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 [26] or with methods that optimize the length of each contact phase in the optimal control problem directly [10].

B. Optimal Control Problem Constraints

With the formulation outlined in Section III-A, 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 [54]:

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

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. (2) can be solved using the Forward Dynamics (FD) [54], 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 $\mathbf{\Phi}$, which we used to formulate the following constraints

$$\mathbf{x}^+ = \mathbf{\Phi}(\mathbf{x}, \mathbf{a}, \mathbf{u}, \boldsymbol{\gamma}, \Delta t), \tag{3}$$

To improve numerical conditioning, the contact point velocity is corrected with a slack variable $\boldsymbol{\gamma}$ as proposed in [45]. This allows to impose 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 [45]. 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 $\mathbf{J}_c\mathbf{v}$:

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

Contact constraints

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

a) 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:

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

b) 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:

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

c) 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 $N_{c,0}$.

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

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

Key-frames 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 kf) need to be imposed in order to not penetrate the ground. This is enforced along the whole optimization horizon through inequalities of the type:

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

Cyclicity

Cyclic motion patterns are the target of the optimization. This choice allows to keep the optimization horizon per cycle short enough without sacrificing numerical precision. Once the motion primitive is obtained, a locomotion pattern that is representative of the robot 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).

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

As these constraints enforce a dependency between the initial and final nodes. A major drawback is that the requirements to define a Markov chain are not respected anymore. This renders using faster iterative algorithms as DDP [38] not viable.

Actuator model and limits

All the main actuator limits are taken into account:

- *Position*: joint position bounds are considered (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. 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 any torque value: the intrinsic limitation due to the bandwidth of the actuation needs to be addressed. Approaches to treating it were proposed in [55], [56] 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 cut-off frequency depends on the actuator technology (for DC motors, it can be estimated $f_c \approx 20Hz$). In time domain, the filter presents a straightforward implementation. For each node $k \in [1..N]$ it results in the following constraints:

$$\begin{cases} \mathbf{u}_k \leq (1-\alpha)\mathbf{u}_{k-1} + \alpha\overline{\mathbf{u}} \\ \mathbf{u}_k \geq (1-\alpha)\mathbf{u}_{k-1} + \alpha\underline{\mathbf{u}} \\ \underline{\mathbf{u}} \leq \mathbf{u}_0 \leq \overline{\mathbf{u}} \end{cases}, \tag{12}$$

where $\alpha \in [0,1]$ depends on f_c and the discretization step Δt as follows:

$$\alpha = \frac{2\pi\Delta t f_c}{2\pi\Delta t f_c + 1}. \tag{13}$$

$\underline{\mathbf{u}}, \overline{\mathbf{u}}$ are respectively the minimum and maximum torque that can be achieved by the actuator. By construction, (12) respects peak limits, as $\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \overline{\mathbf{u}}\ \forall k \in [1..N]$.

C. 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 [26], [27]. P_{el} is computed with the non-ideal dissipations of the actuators.

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

$$P_f = \mathbf{v}_a^\top \underbrace{(\tau_\mu \text{sign}(\mathbf{v}_a) + b\mathbf{v}_a)}_{\tau_f}, \tag{14}$$

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

$$\boldsymbol{\tau} = \mathbf{u} + \boldsymbol{\tau}_f \tag{15}$$

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

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

$$P_t = \boldsymbol{\tau}^\top \mathbf{K} \boldsymbol{\tau}, \tag{16}$$

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

c) Mechanical energy invariance: For any periodic trajectory Ω, the electrical energy equals the integral of the losses ($P_{el} = P_f + P_t$). Therefore, it is not necessary to minimize explicitly the mechanical power $P_m = \mathbf{u}^\top(\gamma)\mathbf{v}_a(\gamma)$ 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):

$$\oint_\Omega \mathbf{u}^\top(\gamma)\mathbf{v}_a(\gamma)d\gamma = (\mathrm{E}_{mec} = \mathrm{E}_{kin} + \mathrm{E}_{pot})|_{\mathbf{x}_0}^{\mathbf{x}_T} \triangleq 0 \text{ as } \mathbf{x}_0 = \mathbf{x}_T \tag{17}$$

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).

IV. Real hardware results

In this section, we present the current quadruped development at DFKI, and we validate the actuator and the power consumption models introduced in Section III-C.

A. The new quadruped prototype at DFKI

The DFKI Robotics Innovation Center recently developed a new robot quadruped (see Fig. 1). The validation and optimization results are based on its preliminary design, which is presented in this section. The robot consists of a central body on which four legs of identical design are mounted. Each leg has 3 degress 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. The physical dimensions for the initial design are similar to the mjbots-quad [57]. 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 1/2. An adjustment of the belt tension can be utilized by a linearly displaceable tensioning pulley. All structural elements of the robot were designed in such a way that they can be manufactured by waterjet cutting. This results in a cost-effective, quickly adaptable robot design. The components of the body consist of carbon Fiber-Reinforced Plastic (in short 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. Even the gearbox pulleys for coupling the knee joint were made by waterjet cutting. The low-cost in-house production by waterjet cutting enables quick replacement of parts and simple adaptation of the kinematics to changed application scenarios or improvements derived by co-design experiments. For example, the leg segments' length or the belt drive's transmission ratio can be adapted very easily. The CAD models were generated in a correspondingly adaptive manner. Lastly, the 3D-printed feet of the robot are exchangeable. This allows different material hardnesses to be tested for different substrates. The material TPU in the Shore hardnesses from 75A to 85D has been used for 3D printing of different feet. In addition, the flexibility of the feet can be adjusted by varying the density of the infill.

B. Actuator model and power consumption validation

The trajectory optimization formulation introduced in Section III is used to produce an energy-optimal bounding motion (for more details on the task, see Section V-B). By tracking the optimal reference trajectory with the prototype, the gap between the model and reality is assessed, and the models are validated. Fig. 3 shows that the actuator model with the identified parameters, closely predicts the total joint torque $\boldsymbol{\tau}$ (including joint friction $\boldsymbol{\tau}_f$) as in (15). 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 [0.3s-0.5s], which can be attributed to the unmodelled controller dynamics. In Fig. 4, 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 $P_{el} = P_{mech} + P_t + P_f$, with the notation introduced in Section III-C. To compute the values, τ is inferred by our joint friction model. Fig. 4 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, $P_{el} = iV$ (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. In addition, a hand-tuned heuristic was used to produce similar jumps with the prototype (with similar time horizon and jump displacement). On this heuristic, the same method was applied to assess the electrical power consumption. It was found that the consumption of the heuristic was higher than the energy optimal trajectories (which are 30% more efficient with respect to the measured values). The optimal energy expenditure is rather accurately estimated by the method, while for the heuristic, the prediction on the reference trajectory overestimates the power consumption.

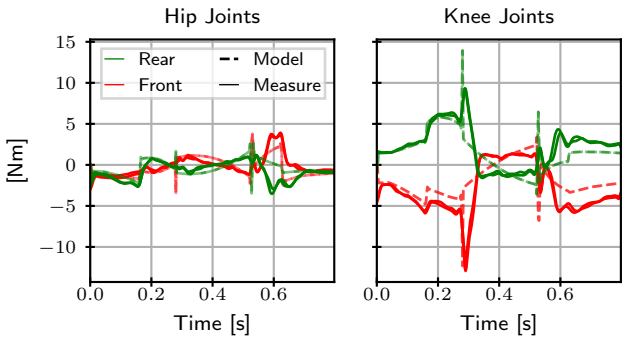


Fig. 3. The actuator model allows a close match between the ideal trajectories with friction compensation and the ideal torque applied to the system from measurement data.

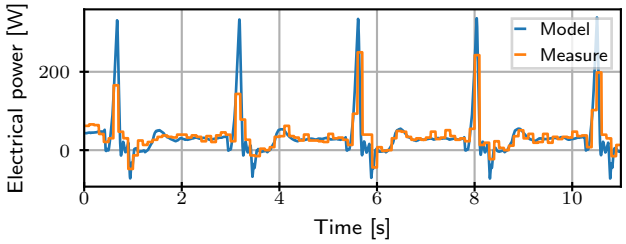


Fig. 4. The electrical power estimation closely follows the values measured on the system.

TABLE III
ENERGY CONSUMPTION VALUES FOR THE JUMP

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

V. CO-DESIGN OPTIMIZATION RESULTS

A. Problem requirements and assumptions

The 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. 5. 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: Fig. 5 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. Another preliminary design choice is to drive both the abduction joint and the hip joint directly, while using a belt transmission with at reduction factor of 2 for the knee joint.

Since the motion of leg abduction in the lateral plane (y,z) is not strictly needed for bounding or jumping, a planar model was used instead of the complete one, to avoid unnecessary complexity. The masses of the motors are lumped on a single axis, and the abduction of the leg (rotation around x of the first leg joint) is blocked. The motor of the blocked DoF is located in the base, while the hip and knee motors are on the same axis and are shown in grey. Nonetheless, the mass of the motors responsible for the leg abduction is considered in the base. This choice simplifies the problem by reducing and coupling some DoFs.

The robot model for this task exploits the symmetry of the motion with respect to the (x,z) plane. The optimization then removes the burden of discovering symmetrical behaviors by encoding them directly in the dynamics. Under these constraints, the dynamical equivalence between the complete 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 limits) is then doubled, and they need to be equally divided into two legs to pass on the real system.

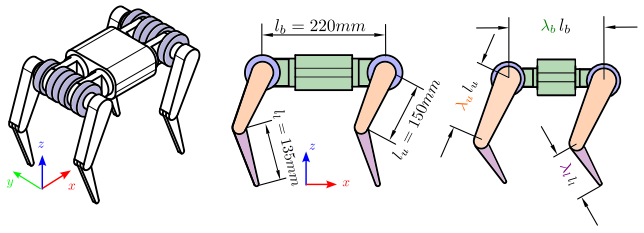


Fig. 5. Complete robot model (left), its planar simplification (center) and scaling of the base, upper leg and lower leg links.

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

Structural scaling of the model

The legs and torso structure are then modeled as fixed payloads, corresponding mainly to the mass of the motors. Additionally, there is a smaller contribution to the mass from embedded electronics and fixating frames for the panels. Taking this into account, 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. 5). For the planar quadruped model of Fig. 5, 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. The mass and dimension of the fixed payload (e.g., motors) do not scale with the rest of the rigid bodies. 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 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 $\propto \lambda^3$.

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:

- 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 specific co-design application the motors are chosen 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 possible motor combinations for the hip and knee motors (respectively m_{hip}, m_{knee}) are then four.

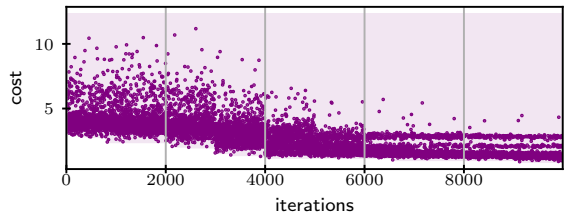


Fig. 6. Convergence of the algorithm along the evolution of the populations.

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:

a) Inertia: The added rotor inertia is considered in the model via the technique explained in Ch. 9.6 of [54]: 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. Moreover, each motor mass contribution is added to the parent link mass and inertia as a concentrated mass.

b) 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 [26].

c) Motor placement: The contribution given by the motor masses is also taken into account by the structural base scaling. Each motor is modeled as a localized mass and the rotational inertia is modified accordingly.

B. 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. 7c):

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

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 0.7 s, and the total number of nodes for the optimization horizon is 100.

a) 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). Fig. 6 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 design for the 4 combinations of the motors.

TABLE V
RESULTS OF THE OPTIMIZATION FOR THE BOUNDING TASK

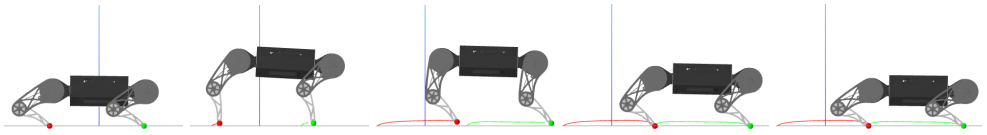
	Nominal	**Optimized**
Cost $\mathcal{L}$	1.78	0.41
λ_u	1.0	0.752
λ_l	1.0	0.514
λ_b	1.0	0.512
m_{hip}	AK80-6	AK80-9
m_{knee}	AK80-6	AK80-9

b) Cost comparison: 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.

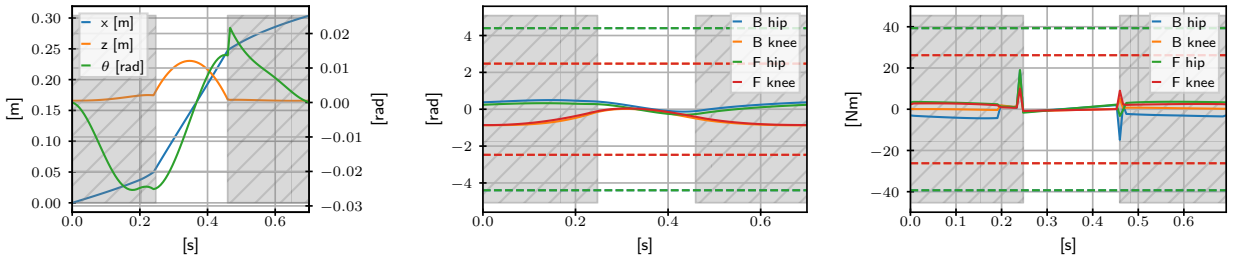
c) Discussion: The method consistently provides results with $|\boldsymbol{\gamma}| < 10^{-6}$. According to Table V, The optimization selects as best suited a smaller robot, with a different scaling of the thigh λ_u and shank λ_l, in particular $\lambda_u/\lambda_l = 1.46$. For jumping forward it seems then that robots with longer thighs are performing better. The optimal solution is chosen so 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. 7b: 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. 7c), the knee stopper can partially limit the robot 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. Basically, 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 is equal to the quotient of the square of their gear ratio, so 2.25. Furthermore, a higher reduction is also impacting the system inertia and reducing the transparency. However, for bounding with the optimal robot (which is smaller), apparently there is no need to use more dynamic and less energy efficient motors, hence a higher gear ratio is selected.

C. Co-design for backflip

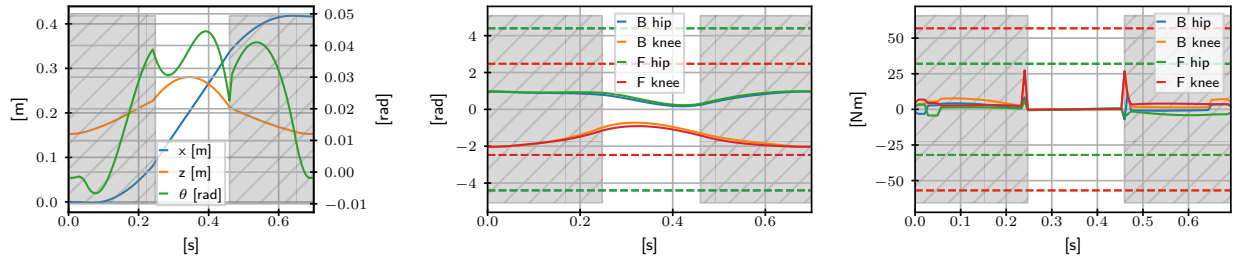
As a second task we present the result of a backflip optimization, as shown in Fig. 8. This motion was selected as a complex and dynamic task example, exploiting the whole-body dynamics of the system. The robot starts with zero velocity and has to perform a full rotation of the base before landing. In the landing phase, the excess velocity needs to be damped to reach a full stop at the end of the trajectory. Moreover, also for this task, all joint positions except the base



(a) Bounding motion

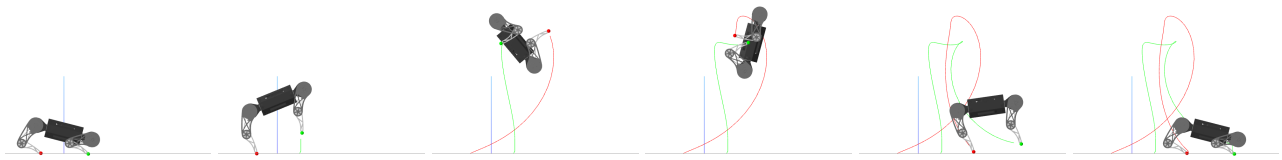


(b) Optimal design minimal energy trajectories

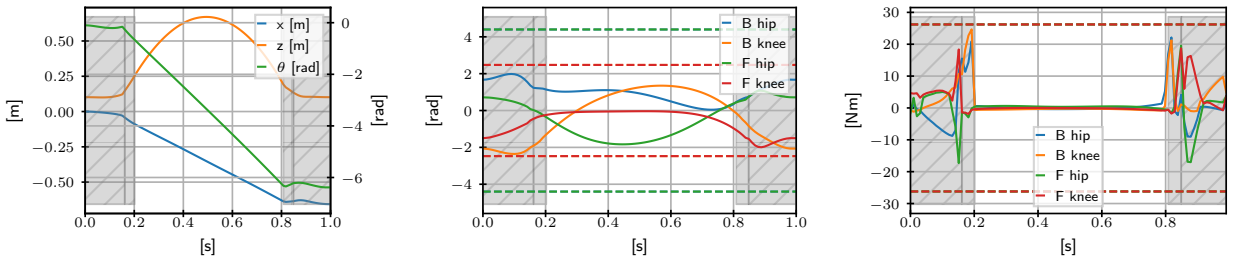


(c) Nominal design minimal energy trajectories

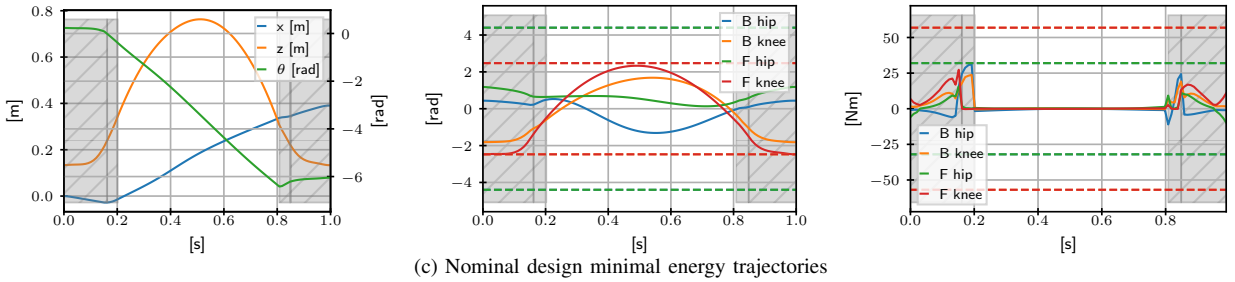
Fig. 7. *Bounding task*: Fig. 7a shows the different motion phases. Trajectories for the optimal and the nominal designs are respectively shown in Fig. 7b and Fig. 7c. In both, from left to right, the plots show: base, joint positions and joint torques trajectories. Contact phases are highlighted with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



(a) 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.



(b) Optimal design minimal energy trajectories



(c) Nominal design minimal energy trajectories

Fig. 8. *Backflip task*: Fig. 8a shows the different motion phases. Trajectories for the optimal and the nominal designs are respectively shown in Fig. 8b and Fig. 8b. In both, from left to right, the plots show: base, joint positions and joint torques trajectories. Contact phases are highlighted with grey background.

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

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. 8a, the different phases are as follows.

- *Dual 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. 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.
- *Dual 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 by 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 design 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.

Cost comparison: The results for 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.

a) Discussion: 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. Fig. 8b and Fig. 8c 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 trade-off 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 trade-off 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. For this task the knee joint is reaching the position limits of the actuator, in both trajectories, so this constitutes another hint that this limit needs to be taken into consideration for the final robot design. Compared to the torque required for bounding, in this case saturation is reached. In Fig. 8b and 8c, 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 the backflip a motor that can produce faster motion is needed for task completion. So, for this task, the motor selection goes in the opposite direction to what was obtained for the bounding task, leading to a smaller reduction to achieve a higher joint velocity. 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.

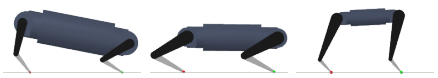


Fig. 9. Center: nominal, left: optimized backflip, right: optimized bounding.

D. Landscape analysis for multiple objectives

As expected, rather different designs were produced for the two tasks by the co-design optimization (Table V and VI, Fig. 9). Therefore, 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 we studied the leg design for the two tasks presented before. The scaling of the upper and lower leg link is then studied 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. 10 and 11 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 frontier was reconstructed (Fig. 12) for the two different task costs. The resulting Pareto front is reported in Table VII, 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

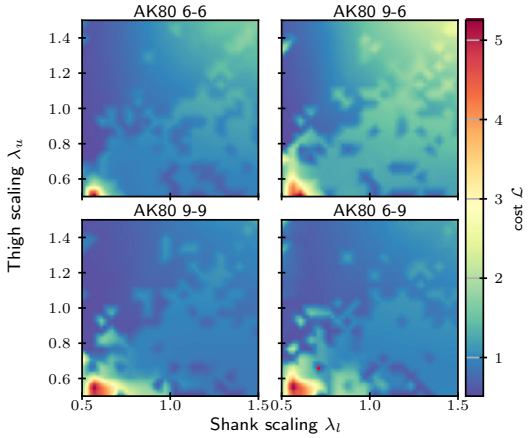


Fig. 10. Jumping task landscape for motor and link scaling combinations.

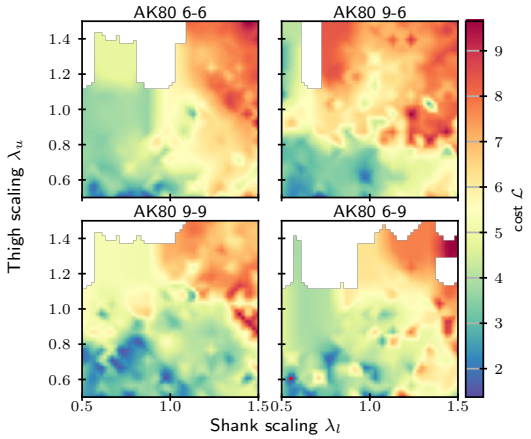


Fig. 11. Backflip task landscape for the different motor combinations and link scaling. The white regions are associated to unfeasible problems, which occur consistently when the robots have short shanks and long thighs.

the main movement mode). So the closest options are the 2nd and 3rd rows of Table VII, 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. 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.

TABLE VII
PARETO OPTIMAL DESIGNS FOR THE TWO TASKS

λ_u	λ_l	m_{hip}	m_{knee}	$\mathcal{L}$ **Backflip**	$\mathcal{L}$ **Bounding**
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

Lower values of $\mathcal{L}$ indicate better performance.

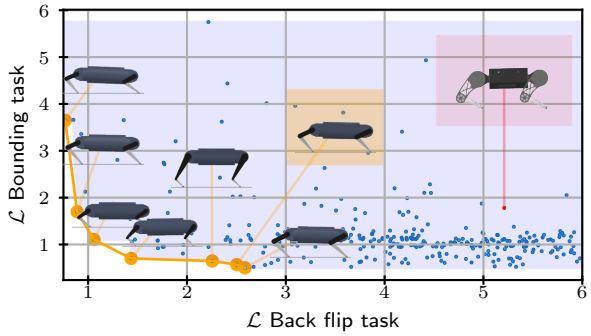


Fig. 12. Pareto front approximation for the two tasks' cost. The designs are superimposed. The one highlighted in orange is the design which is requires the least modifications to the nominal prototype, shown in red.

VI. CONCLUSIONS AND FUTURE WORK

In this work, our co-design framework was improved to gain completeness and versatility, and we applied it to the optimization of a quadruped robot developed at DFKI-RIC. Two cyclic tasks were selected to represent different key motion capabilities. In the initial phase, optimal hardware solutions were found for each of the two tasks separately. Since the two optimizations led to rather different designs, we used a Pareto set approach to select a versatile and efficient trade-off. This insight will be used for the development of the next quadruped version. The core contributions of the proposed co-design framework are: first, the development of a more robust and parallelizable bi-level scheme capable of handling mixed-integer variables and, second, a more versatile OCP formulation with equality and inequality constraints, which allows coping with actuation bandwidth and motion cyclicity. The energy consumption and friction models of the actuator used in trajectory optimization were validated on the current prototype robot. In future work, we plan to include optimal gain selection in the OCP. However, the main limitation of the method remains the computational cost of the OCP resolution, which may become a bottleneck for more complex systems or tasks. Finally, even though key information is provided by this tool, expert knowledge is still needed both to select a design on the Pareto set and to practically implement it on hardware.

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.

REFERENCES

[1] P. Wensing, "The rapid rise of quadruped robots [young professionals]," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, 2021.
[2] Anybotics, *Anymal X*, https://www.anybotics.com/anymal-x-the-worlds-first-ex-proof-legged-robot/.
[3] B. Dynamics, *Spot specifications*, https://support.bostondynamics.com/s/article/Robot-specifications.
[4] M. Hutter, C. Gehring, D. Jud, *et al.*, "ANYmal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[5] 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," *The International Journal of Robotics Research*, vol. 33, no. 11, 2014.

[6] 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*, vol. 36, no. 2, 2017.

[7] 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 Int. Conf. on Intelligent Robots and Systems*, 2018.

[8] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *IEEE Int. Conf. on Robotics and Automation*, 2019.

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

[10] K. D. Mombaur, R. W. Longman, H. G. Bock, and J. P. Schlöder, "Open-loop stable running," *Robotica*, vol. 23, no. 1, 2005.

[11] G. Saurel, J. Carpentier, N. Mansard, and J.-P. Laumond, "A simulation framework for simultaneous design and control of passivity based walkers," in *IEEE Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, 2016.

[12] 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 Int. Conf. on Intelligent Robots & Systems*, 2017.

[13] S. Coros, B. Thomaszewski, G. Noris, *et al.*, "Computational design of mechanical characters," *ACM Trans. on Graphics*, vol. 32-(4), 2013.

[14] 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," *Mobile Service Robotics*, pp. 315–323, 2014.

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

[16] Y. Yesilevskiy, Z. Gan, and C. David Remy, "Energy-optimal hopping in parallel and series elastic one-dimensional monopeds," *Journal of Mechanisms and Robotics*, vol. 10, no. 3, 2018.

[17] Y. Yesilevskiy, Z. Gan, and C. Remy, "Optimal configuration of series and parallel elasticity in a 2D monoped," *IEEE Int. Conf. on Robotics & Automation*, 2016.

[18] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *Int. Journal of Robotics Research*, vol. 37, 2018.

[19] 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*, 2020.

[20] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "A versatile co-design approach for dynamic legged robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2022.

[21] G. Bravo-Palacios and P. M. Wensing, "Large-scale ADMM-based co-design of legged robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2022.

[22] 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.

[23] J.-H. Park and H. Asada, "Concurrent design optimization of mechanical structure and control for high speed robots," *Journal of Dynamic Systems, Measurement, and Control*, vol. 116, no. 3, Sep. 1, 1994.

[24] J. A. Reyer and P. Y. Papalambros, "Combined Optimal Design and Control With Application to an Electric DC Motor," *Journal of Mechanical Design*, vol. 124, 2002.

[25] 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 Int. Conf. on Intelligent Robots and Systems*, 2022.

[26] G. Fadini, T. Flayols, A. Del Prete, N. Mansard, and P. Souères, "Computational design of energy-efficient legged robots: Optimizing for size and actuators," in *IEEE Int. Conf. on Robotics & Automation*, 2021.

[27] G. Fadini, T. Flayols, A. Del Prete, and P. Souères, "Simulation aided co-design for robust robot optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, Oct. 2022.

[28] 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.

[29] 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 Transactions on Robotics*, vol. 33, no. 3, Jun. 2017.

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

[31] 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*, 2021.

[32] 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," *The International Journal of Robotics Research*, vol. 32, no. 8, Jul. 2013.

[33] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv:1604.00772*, 2016.

[34] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, Mar. 1, 2006.

[35] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11-(1), 2019.

[36] J. Carpentier and N. Mansard, "Analytical Derivatives of Rigid Body Dynamics Algorithms," in *Robotics: Science and Systems*, 2018.

[37] J. Carpentier, G. Saurel, G. Buondonno, *et al.*, "The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE/SICE Int. Symposium on System Integration*, 2019.

[38] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, Jan. 1966.

[39] I. S. Duff, "MA57 a code for the solution of sparse symmetric definite and indefinite systems," *ACM Trans. Math. Softw.*, vol. 30, no. 2, pp. 118–144, 2004.

[40] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM Jrnl. on Optimization*, vol. 12, no. 3, 2002.

[41] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IFAC Proceedings Volumes*, vol. 41, no. 2, 2008, 17th IFAC World Congress.

[42] E. Todorov, "A convex, smooth and invertible contact model for trajectory optimization," in *IEEE Int. Conf. on Robotics & Automation*, 2011.

[43] I. Mordatch, E. Todorov, and Z. Popovic, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics*, vol. 31, 2012.

[44] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," in *Algorithmic Foundations of Robotics X*, Springer, 2013.

[45] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *IEEE Int. Conf. on Robotics & Automation*, 2016.

[46] S. Kajita, F. Kanehiro, K. Kaneko, *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE Int. Conf. on Robotics & Automation*, vol. 2, 2003.

[47] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa, "ZMP-based biped running control," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, 2007.

[48] S. Kajita, M. Morisawa, K. Miura, *et al.*, "Biped walking stabilization based on linear inverted pendulum tracking," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[49] 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.

[50] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2, Oct. 2013.

[51] 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.

[52] K. H. Koch, K. Mombaur, and P. Souères, "Optimization-based walking generation for humanoid robot," *IFAC Proceedings Volumes*, vol. 45, no. 22, 2012, 10th IFAC Symposium on Robot Control.

[53] C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica*, no. 5, 2001.

[54] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[55] N. K. Gupta, "Frequency-shaped cost functionals - extension of linear-quadratic-gaussian design methods," *Jrnl. of Guidance Control and Dynamics*, vol. 3, 1980.

[56] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for torque-controlled legged robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019.

[57] J. Pieper, *Mjbots quad*, https://github.com/mjbots/quad.