

Optimal Design of Agile Jumping Maneuvers for a Single Leg System

Andrea Patrizi^{*†}, Francesco Ruscelli^{*‡}, Arturo Laurenzi^{*‡}, and Nikos G. Tsagarakis^{*}

Abstract—In this work we showcase a comprehensive study, developed around our heavy-duty electrically-actuated quadruped leg prototype, focusing on the task of jumping. Our study employs a two-stage pipeline for the design and execution of agile and efficient jumping maneuvers. We combine offline non-linear whole-body Trajectory Optimization (TO) to formulate and solve two distinct non-linear problems, each tackling different phases and performance aspects of the jumping maneuver. The first one jointly addresses the thrust and flight phases and aims at maximizing the achieved jump height, while accounting for several hardware limitations. The second problem deals with the impact and braking phases minimizing the kinetic energy dissipated at the impact, while maximizing the energy regenerated by the actuators during braking. We validate the accuracy of the employed models and the results of the optimizations both in simulation and on the real leg hardware, showing the effectiveness of our approach in maximizing hardware performance and the benefits of exploiting combined impact-energy considerations during the execution of jumping maneuvers.

I. INTRODUCTION

Legged robots are among the most versatile robotic platforms available. As their natural counterpart, they show great flexibility and are particularly suited for applications where navigation through unstructured environments is required.

The development of robots capable of fully exploiting the potential of legged morphology is, however, a non-trivial task, both in terms of design and control. There are several instances of robots that come close to achieving this. The autonomous quadruped ANYmal [1] is, for instance, an example of a torque-controllable robot capable of robustly navigating through harsh industrial environments while performing complex locomotion tasks.

The potential of legged robots has begun to show its full realization only in recent years, culminating with the latest Boston Dynamics’ ATLAS [2] performance, where human-like agile and complex jumping maneuvers were demonstrated on real hardware. The success of Boston Dynamics’ approach shows how the convergence of both high-level hardware and carefully designed planning and control software is a critical requirement for future state-of-the-art robots.

Motivated by the end goal of developing a new electrically powered agile quadruped, we present an in-depth case study revolving around our heavy-duty torque-controllable 12 kg quadruped leg prototype (shown in Fig. 1). We choose to focus our attention on the most meaningful and complex task

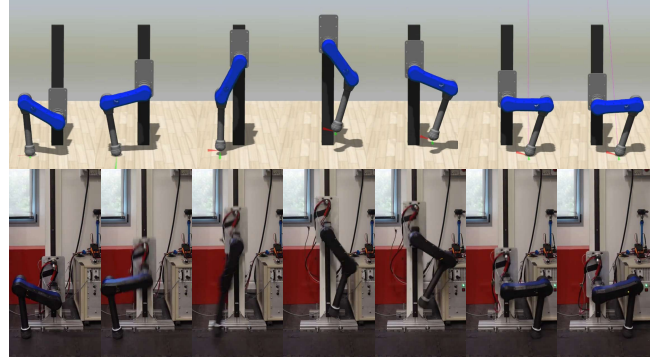


Fig. 1. Snapshots of our quadruped leg prototype performing the optimized jump and landing maneuver obtained with our two-stage TO formulation, in simulation (top) and on the real hardware (bottom), from left to right.

achievable with a single leg setup: the jump. To be able to perform a jumping task while fully exploiting the capabilities of our prototype, we developed a two-stage pipeline (shown in Fig. 2) based on Trajectory Optimization (TO), which in recent years has become one of the most successful methods in generating complex motions for robotic systems [3]–[5].

The core contribution of this work is the development of a two-stage TO pipeline for designing and executing jumping maneuvers, while maximizing energy regeneration and impact mitigation, incorporating also accurate actuation and power flow models. In literature, there exists several works reasoning about impact mitigation and energy recuperation, e.g. [4], [6], [7]. However, to the authors’ best knowledge, this is the first time the relationship between ground impact and energy regeneration is explored and accounted for *explicitly* in the context of jump maneuver design for dq0-based field-oriented-controlled robotic systems. The validation of our two TOs and models was performed through extensive trials in simulation and on the real hardware.

The rest of the paper is organized as follows. Section II lays down the premises for our approach and TOs. Section III details the formulation of our models and optimization problems, focusing on some pivotal aspects specific to the task under analysis, i.e. the jump. Section IV shows the results of our study providing experimental validation for the TO formulations. Finally, Section V sums up and comments on the main results of our case study and highlights possible lines of future research.

II. PROBLEM FRAMING AND STATE OF THE ART

A jumping task may be split into four main phases (shown in Fig. 2): the take-off or thrust, the flight, the impact and the braking. We start by exploring the phases individually

[†]Department of Informatics, Bioengineering, Robotics and Systems Engineering, Università di Genova, Via All’Opera Pia 13, 16145 Genova.

^{*}Humanoids and Human-Centred Mechatronics (HHCM), Istituto Italiano di Tecnologia (IIT), Via San Quirico 19d, 16163 Genova.

[‡]The authors contributed equally to the manuscript.

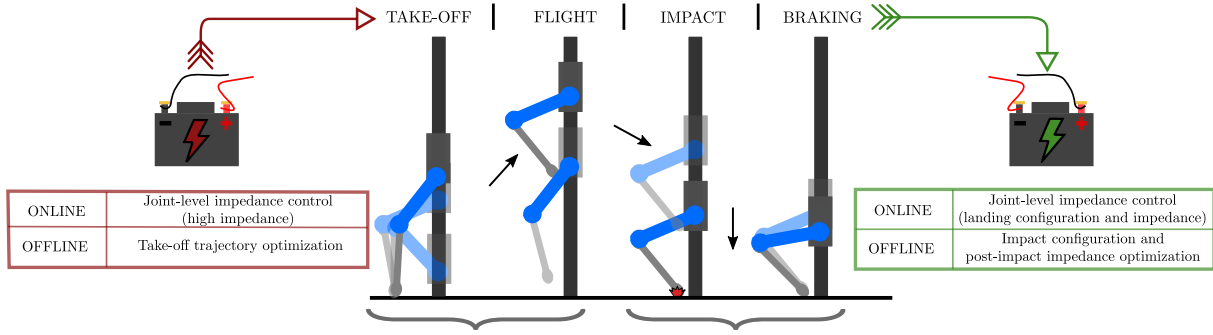


Fig. 2. A jump sequence decomposed into four main phases, which are synthetically depicted at the center of the figure: the takeoff, the flight, the impact and the braking. During the takeoff phase, energy flows from the power source towards the actuators; after the impact, however, the leg brakes and some of the residual kinetic energy is sent back to the power source and is thus available for regeneration.

and assess the challenges and peculiarities associated with each of them.

During the take-off phase the robot has to accelerate its center of mass (CoM) sufficiently to break the contact from the ground, while modulating the contact forces exchanged with the environment. At the impact and while braking, an amount of the available kinetic energy, on the basis of the properties of the colliding bodies (foot and ground) and the configuration of the robot at the impact instant [8], may be lost. During the post-impact braking phase, any residual kinetic energy, if available, needs to be somehow dissipated in order to brake the robot. Thus, the question of whether this energy can be harvested or not, instead of being lost in the robot structure and the environment, rises. Looking at nature, it has been highlighted how biological jumpers tend to employ energy storage principles during the landing phases to be able to increase the performance and efficiency of their movements [9], [10]. In the case of our prototype (powered by torque controlled actuators similar to those also used on CENTAURO [11] robot), it is possible to exploit a similar strategy and make use of the power source in place of tendons as temporary energy storage of mechanical energy by performing what is called *energy regeneration* [12], [13] during braking phases.

Based on these considerations, we choose to approach the problem of generating a jumping motion for our prototype in a modular way. Instead of solving a single optimization problem for the whole jump horizon, we split the problem into two separate ones (as shown in Fig. 2) to gain simplicity both in terms of formulation and computational complexity. We tackle the take-off and flight phases (up to the apex) in the first one, with the objective of maximizing the apex height, while we address the remaining phases in the second. Given the characteristics of the torque controlled actuation used to power our robot leg we employ a joint space impedance controller to track the references provided by our take-off TO. For the landing phase, we restrict our attention to a specific braking strategy: after the apex, the joint level impedance controller is fed with desired landing impedance set points and a fixed reference configuration. The main idea for the braking phase is to see if it's possible to optimize the impedance gains and configuration to maximize the recovered energy while also preventing high impacts.

III. FORMULATION

A. System modeling

The dynamics of a general floating-base system can be written as

$$B(q)\dot{v} + C(q, v)v + g(q) = \tau_a + J^T f \quad (1)$$

where $B \in \mathbb{R}^{n_v \times n_v}$ and $C \in \mathbb{R}^{n_v \times n_v}$ are, respectively, the generalized inertia and Coriolis matrices of the system; $q \in \mathbb{R}^{n_q}$ and $v \in \mathbb{R}^{n_v}$ are the configuration and generalized velocity vectors of the system, while $g \in \mathbb{R}^{n_v}$ and $\tau_a = [0_{6 \times 1}; \tau_m] \in \mathbb{R}^{n_v}$ are, respectively, the generalized gravity vector and the actuation torques. Finally, $J^T(q)f$ are the torques generated by the generalize wrench f through the contact jacobian J . In the specific case of our leg prototype, the system has a degenerate floating base with only a single degree of freedom, given by the sliding guide joint (see Fig. 1); as a consequence, τ_a is simply $[0; \tau_m]$, $n_q = n_v = 3$, $J \in \mathbb{R}^{3 \times 3}$ and $f \in \mathbb{R}^3$, if we only consider a point contact with the ground. Additionally, the first component q_0 of q represents the vertical position of the floating joint, and the remaining components q_1, q_2 represent, respectively, the hip and knee joint positions.

B. Actuator modeling: friction-aware quadrature current estimation

It is known [14] that the torque-quadrature current characteristic for Brushless Direct Current (BLDC) actuators is well approximated by the relationship

$$\tau_m = K_t i_q \quad (2)$$

with the $K_t \in \mathbb{R}$ is the torque constant of the motors. Employing this information, we can write a friction-compensated version of the rotor-side dynamics as in [15]:

$$K_t i_q - \tau_l \eta + \tau_{fl} \eta = I_r \frac{\dot{v}_l}{\eta} \quad (3)$$

where $\tau_l \in \mathbb{R}^2$ are the link side torques acting on the actuators, $\tau_{fl} \in \mathbb{R}^2$ is a vector modeling the friction torques reflected at the link-side, $i_q \in \mathbb{R}^2$ are the quadrature currents, $0 < \eta \leq 1$ is the reduction ratio from the motor to link side and I_r is the axial inertial of the rotor. Equation (3) can be used to build a friction observer (as done in [15]) or,

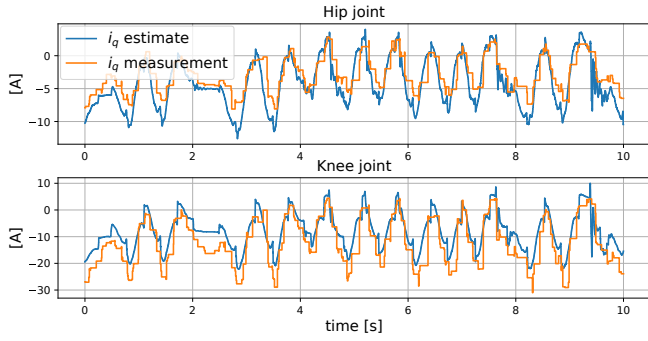


Fig. 3. Validation of the quadrature current i_q model during a simple test: the leg stands on the ground while using relatively low-joint impedance and a vertical pushing force is applied to the base link. The tracking of the calibrated model is accurate, thus justifying its incorporation into our TO formulation.

alternatively, can be employed to calibrate a suitable friction model for τ_{fl} . Specifically, we use the simple Coulomb-like friction model

$$\tau_{fl} = -K_{f,s} \text{signh}(v_l) - K_{f,d} v_l \quad (4)$$

where $\text{signh}()$ is a smooth approximation of the sign function based on the hyperbolic tangent and the scalars $K_{f,s}$ and $K_{f,d}$ are the static and dynamic friction coefficients respectively. We employ (3) to estimate the friction coefficients in (4) by solving a simple least-squares regression problem with data collected during suitable calibration trajectories. Figure 3 shows the tracking performance of the resulting calibrated model w.r.t. actual measurements.

C. Take-off optimization: maximizing the reached apex height

Given the scope of our case study, we are interested in performing the jumping task with the highest performance our prototype is able to offer. For a jumping trajectory, this means maximizing the height at the apex of the jump, while accounting for the hardware's constraints. To achieve this, we employ an inverse dynamics-based formulation and hence use accelerations \ddot{v} and contact forces f as inputs to the rigid body dynamics (1). The benefits of employing inverse-dynamics based formulations, as opposed to forward-dynamics based ones, are starting to become apparent [16], [17]: an increased numerical efficiency and stability, particularly for coarse discretizations. We define the states and inputs vectors x_t and u_t of our take-off TO as

$$x_t = [q, v, \dot{v}, f] \quad (5)$$

$$u_t = [\ddot{v}, \dot{f}] \quad (6)$$

The running cost of the TO is formulated as

$$\ell_t(x_t, u_t) = \ell_f + \ell_{\dot{f}} + \ell_v + \ell_{\dot{v}} + \ell_{\ddot{v}} \quad (7)$$

where the *weighted* costs $\ell_f, \ell_{\dot{f}}, \ell_v, \ell_{\dot{v}}, \ell_{\ddot{v}}$ are all quadratic in, respectively, f , the yank \dot{f} , v , \dot{v} and the jerk \ddot{v} . The cost for the maximization of the CoM height h_{CoM} is implemented as a terminal constraint at the apex time T_{apex} of the form

$$\ell_{\text{CoM}}^F(T_{\text{apex}}) = \frac{1}{h_{\text{CoM}}(q(T_{\text{apex}})) + \varepsilon} \quad (8)$$

where ε is a small positive number to avoid singularities. The only non-linear cost is the term ℓ_{CoM} , which maximizes the height of the CoM. The integral cost of the take-off TO can hence be written as

$$L_t(W_t) = \ell_{\text{CoM}}^F + \int_0^{T_{\text{apex}}} \ell_t(\tau) d\tau \quad (9)$$

where

$$W_t := [x_t, u_t, T_{\text{takeoff}}, T_{\text{apex}}] \quad (10)$$

collects our optimization variables and T_{takeoff} and T_{apex} represent, respectively, the take-off and apex instant. The cost function (9) is complemented by a set constraints, which are used to enforce both the jumping maneuver and a series of crucial hardware constraints. First, defining τ_{lim} as the actuated joints torque limits, we impose bounds on the torques τ_a :

$$-\begin{bmatrix} 0 \\ \tau_{\text{lim}} \end{bmatrix} \leq \tau_a \leq \begin{bmatrix} 0 \\ \tau_{\text{lim}} \end{bmatrix} \quad (11)$$

Furthermore we constrain the tip to stay on the ground before the takeoff instant with the following couple of constraints, respectively, on the tip position χ_{tip} and velocity $\dot{\chi}_{\text{tip}}$

$$\chi_{\text{tip},z} = 0; t = 0 \quad (12)$$

$$\dot{\chi}_{\text{tip}} = 0; t \in [0, T_{\text{takeoff}}] \quad (13)$$

We also impose the apex at the final instant with the terminal constraint

$$\dot{\chi}_{\text{CoM},z} = 0; t = T_{\text{apex}} \quad (14)$$

We impose contact and friction constraints on f with

$$f \geq 0; t \in [0, T_{\text{takeoff}}] \quad (15)$$

$$f = 0; t \in [T_{\text{takeoff}}, T_{\text{apex}}] \quad (16)$$

$$-\mu f_n \leq f_t \leq \mu f_n; t \in [0, T_{\text{takeoff}}] \quad (17)$$

At the first instant, the leg should also start still; as a consequence

$$v = 0; t = 0 \quad (18)$$

We also account for the calibrated actuation model (3) by constraining the quadrature current i_q of the actuator with

$$\text{rotor dynamics (3)} \quad (19)$$

$$-i_{q,\text{max}} \leq i_q \leq i_{q,\text{max}} \quad (20)$$

Finally, we can write the continuous version of our take-off TO as

$$\min_{W_t} L_t(W_t) \quad (21a)$$

subject to

$$\text{rigid body dynamics (1)} \quad (21b)$$

$$\text{constraints (11) to (20)} \quad (21c)$$

$$\text{joint position and velocity ranges} \quad (21d)$$

The continuous TO problem described by 21 is then transcribed over a grid of N nodes by employing the multiple shooting method [18]. The solution to the discretized version of (21) is first re-sampled at a desired constant rate following the procedure detailed in [19] and then fed as

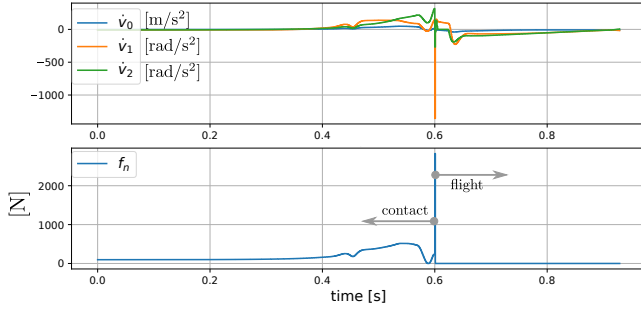


Fig. 4. Joint accelerations (on top) and normal component of the ground reaction force obtained for a refined take-off problem without jerk and yank minimization: the solver exploits a caveat in the inverse dynamics formulation and finds a solution which respects node constraints, but is not feasible in reality.

initial guess to a *refined* discretized problem with the same costs and constraints as (21), but with fixed time intervals. This procedure, very similar to the “mesh refinement” employed in [19], is needed to make the re-sampled solution dynamically feasible. As shown in [19], in fact, the re-sampling procedure may introduce large dynamics defects on the floating-base of the robot, thus making the obtained solution dynamically unfeasible. As an additional remark, the authors would like to underline the importance of the terms $\ell_{\dot{v}}$ and $\ell_{\ddot{v}}$ in minimizing the feasibility gap between simulation and reality. Fig. 4 shows the results of a refined solution where the regularization costs on the jerk \dot{v} and the yank \ddot{v} are not employed. Looking at the picture, the solver exploits a caveat in the inverse dynamics-based formulation and produces an impulse of f at the takeoff-instant without violating the inverse dynamics on the nodes. It does so by compensating with corresponding impulses of \dot{v} . This acceleration and ground reaction forces are however not feasible in reality, and this could be checked by looking at the dynamics residuals obtained by re-sampling the refined trajectory at a higher rate. This shows how with the inverse dynamics-based formulation applied to highly dynamic TO needs to be complemented with suitable regularization of the jerk \dot{v} and \ddot{v} to function properly.

D. Landing phase: impact dynamics modeling

After the apex of the flight phase, the leg will make contact with the ground. Under the usual assumption that the impact duration is negligible with respect to the time scale over which the body dynamics evolves, it is possible to integrate (1) over the impact instant to obtain the so called *impact dynamics* [20]:

$$B(\hat{q}) [v_+ - v_-] = J^T(\hat{q}) \hat{f} \quad (22)$$

where \hat{q} is the impact configuration of the robot, v_+ and v_- are the post-impact and pre-impact generalized velocities of the system and \hat{f} is the impact impulse, defined as the integral of the impact force during the impact interval. The forward kinematics continues to be valid at the impact instant, so we can write

$$J(v_+ - v_-) = \dot{\chi}_{\text{tip}}^+ - \dot{\chi}_{\text{tip}}^- \quad (23)$$

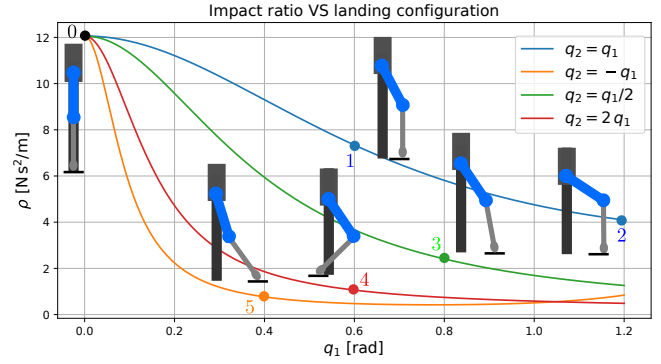


Fig. 5. Depending on the landing configuration, the impact felt by the robot at the ground can vary greatly. In this figure, we show how the modulus of the impact ratio ρ changes with the landing configuration, which is described by the two joint variables q_1 (hip angle) and q_2 (knee angle). We specifically highlight 6 representative configurations, which are shown on top of the impact curves.

where the explicit dependence upon \hat{q} is dropped for simplicity and $\dot{\chi}_{\text{tip}}$ is the cartesian velocity of the end-effector. Looking at equations (22), (23) we see that the result of the impulse \hat{f} is to produce a step in the joint velocities v which, in turn, will produce a step in the end-effector velocity.

Let us look at the energetic aspect of the impact, which is particularly relevant for our case study. The variation of kinetic energy right before and after the impact is given by the simple analytical relationship

$$\begin{aligned} \Delta E_k &= E_k^+ - E_k^- \\ &= \frac{1}{2} (v^{+T} B v^+ - v^{-T} B v^-) \end{aligned} \quad (24)$$

From (22), exploiting the non-singularity of B , we can write

$$v^+ = B^{-1} J^T \hat{f} + v^- \quad (25)$$

where we recall that B is the generalized inertia matrix of the system (1). Substituting (25) into (24) and (23), after a little rearranging, we can obtain the alternative expression

$$\Delta E_k = \frac{1}{2} \hat{f}^T (\dot{\chi}_{\text{tip}}^+ + \dot{\chi}_{\text{tip}}^-) \quad (26)$$

From (24) it is clear that, in general, energy is not conserved at the impact. As a simplification, we assume an inelastic impact with the ground and the velocity of the actuated joints to be negligible. This second assumption was successfully verified both in simulation and on the hardware over several jumping tests. With these premises, we can write a simplified version of (23) and (24) for our specific case as

$$J v^+ - J v_*^- = -\dot{\chi}_{\text{tip}}^- \quad (27)$$

and

$$\Delta E_k = \frac{1}{2} \hat{f}^T \dot{\chi}_{\text{tip}}^- \quad (28)$$

with

$$\dot{\chi}_{\text{tip}}^- = [0, 0, v_{\text{fall}}] \quad (29)$$

and

$$v_*^- = [v_{\text{fall}}, 0, 0] \quad (30)$$

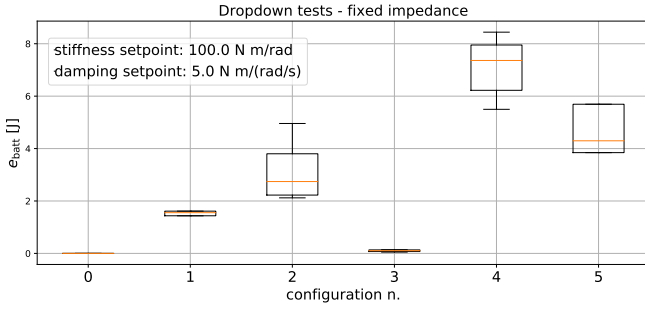


Fig. 6. Results of the dropdown tests performed in simulation, with fixed joint impedance. The leg is dropped from an height (of the tip w.r.t. to the ground) of 0.5 m multiple times with 6 different touchdown configurations chosen manually on the curve 5 (the same shown in Fig. 5). At each touchdown, the regenerative energy flow e_{batt} towards the power source is monitored using the model (34). The results indicate that not all configurations with low impact are equivalently suitable for energy recuperation.

where v_{fall} is the vertical component of the velocity vector right before the impact. Furthermore, we can also speculate, being the fall vertical, that most of the impulse will be concentrated on the vertical component of \hat{f} . As a consequence,

$$\hat{f} = \begin{bmatrix} 0 \\ 0 \\ \hat{f}_n \end{bmatrix} \quad (31)$$

and we can therefore conclude that, since $v_{\text{fall}} \leq 0$ and $\hat{f}_n > 0$ for a vertical fall, $\Delta E_k \leq 0$. Under our assumptions of inelastic and vertical impact, it is possible to combine the impact equations and arrive, after a bit of manipulation, to the following relationship between the normal component of the impact impulse and the pre-impact velocity [21]:

$$\rho := \frac{\hat{f}_n}{v_{\text{fall}}} = -\frac{1}{\Lambda_{2,2}^{-1}(\hat{q}_1, \hat{q}_2)} \quad (32)$$

to which, from now on, we will refer to as the “impact ratio”. $\Lambda_{2,2}^{-1}$ is the third element on the diagonal of the inverse cartesian inertia matrix $\Lambda^{-1} = JB^{-1}J^T \in \mathbb{R}^{3 \times 3}$. For our case in which the floating joint can only move vertically, ρ only depends on the configuration of the leg at the impact $\hat{q}_{12} := [\hat{q}_1, \hat{q}_2]$. Since we cannot know exactly when the impact will happen, v_{fall} cannot be used to reduce the dissipation of kinetic energy. The only option is to reduce \hat{f} or, equivalently, ρ , which we know will depend on the landing configuration. Exploiting the simplicity of (32), we can have a quantitative glimpse at how the impact varies depending on the landing configuration. This is exactly what is shown in Fig. 5. Most notably, the configuration n.0 is the absolute worst in terms of impact: all the kinetic energy is dissipated at the impact through the ground and, more importantly, through the robot itself. As a consequence, without choosing properly the landing configuration, we could potentially endanger the mechanical integrity of the robot and drastically reduce the energy available for regeneration during the braking phase.

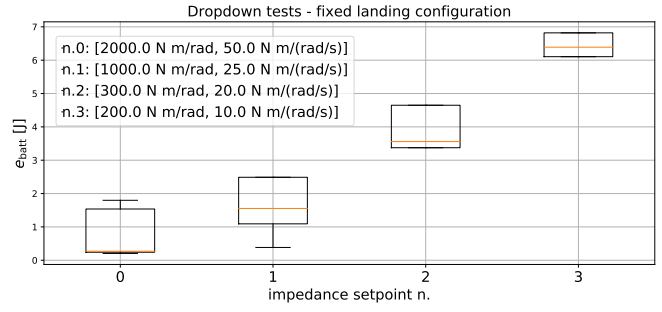


Fig. 7. Results of the drop-down tests performed in simulation, with the landing configuration fixed to the n. 4 (shown in Fig. 5). The leg is dropped from an height (of the tip w.r.t. to the ground) of 0.5 m multiple times with 4 different joint impedance setpoints (same for all actuated joints), shown at the top left of the figure (stiffness and damping set points, respectively). At each touchdown, the regenerative energy flow e_{batt} towards the power source is monitored using the energy model (34). The results confirm that landing with stiff joints greatly diminishes the amount of energy recovered. On the contrary landing with “soft” joints diminishes the joule losses inside the actuators, due to the lower required braking torques.

E. Landing phase: energy regeneration

In Section II we have already qualitatively highlighted how during the braking phase it is possible to recover some of the post-impact residual energy. Here we outline the formulation of a simple and effective model of the energy flow from the power source (in our case a battery) and the actuators, which will be employed in the braking TO problem outlined in the upcoming Section III-F.

When performing Field Oriented Control on BLDC motors (like in our case), it is possible to write the power balance from the battery towards the actuators in the so called $qd0$ reference frame employing the Park transform [14] as

$$p_{\text{batt}} = -\frac{3}{2} \left(R i_q^2 + L_q i_q \frac{d}{dt} i_q \right) - K_t i_q v_m \quad (33)$$

where R and L_q are the phase resistance and quadrature inductance of the motors and v_m is the rotor velocity. The relationship, for simplicity, is shown for a single actuator. Equation (33) can be integrated over an arbitrary interval of time to obtain the energy flow as

$$e_{\text{batt}}(t) = e_{\text{batt}}(t_0) - \frac{3}{2} R \int_{t_0}^t i_q^2 d\tau - \frac{3}{4} L_q [i_q^2(t) - i_q^2(t_0)] - K_t \int_{t_0}^t i_q v_m d\tau \quad (34)$$

Inside (34) we can distinguish, in this order, the initial energy level of the battery, the dissipative joule losses, a conservative term representing the energy stored inside the motor phases and the mechanical energy at the motor rotor. For our purposes, the inductive term in (34), which is conservative and also weighted by L_q (usually in the order of μH), can be neglected.

Figures 6 and 7 report data from two series of drop-down tests performed in simulation while monitoring the energy flow with (34). After each touchdown, the energy recovered is evaluated using (34). The results show that not all low-impact configurations are suitable for energy regeneration and that the amount of recovered energy is indeed also

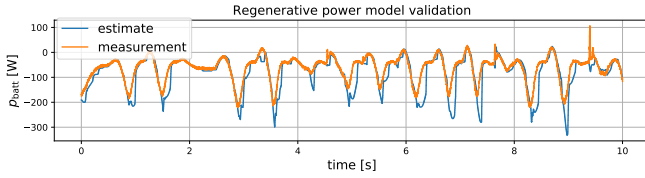


Fig. 8. Validation of the model of the power flow model (33): the leg stands on the ground with low-joint impedance and an vertical oscillating force is applied to the base link. Looking at the plot, we can confirm that the model is indeed accurate in predicting the power flow of our setup.

influenced greatly by the employed joint impedance set points. The model (33) was also successfully validated on the real hardware by employing a sensing setup made of a custom data acquisition system which allows to monitor the voltage and current on the power bus. The results of the validation are shown and briefly discusses in Fig. 8.

F. Post-impact optimization: minimizing impact impulse and maximizing energy recovery

Motivated by the results shown in Fig 7 and 6 and 5, we formulate a suitable TO to jointly address the impact and power regeneration during the braking phases. As done in Sec III-C we employ an inverse dynamics-based formulation. In this case, the running cost takes the form

$$\ell_b(x_b, u_b) = \ell_{p_{\text{batt}}} + \ell_f + \ell_{\dot{v}} \quad (35)$$

where

$$x_b = [q, v] \quad (36)$$

$$u_b = [\dot{v}, f] \quad (37)$$

and ℓ_f and $\ell_{\dot{v}}$ are simple *weighted* quadratic regularizations on the inputs. In this case, being the braking a less dynamic phase than the take-off, we don't employ the additional regularization terms on the \ddot{v} and \dot{f} (see Sec. III-C). The weighted stage cost $\ell_{p_{\text{batt}}}$ is build using directly $-p_{\text{batt}}$ (33) and neglecting the inductive term. Differently from before, we formulate the problem with a fixed phase duration T_f and impose the impact dynamics laws of Section III-D only at the initial instant of the optimization horizon, where v^- (22) assumes the role of a parameter of the problem. To encourage impact mitigation, we use a *weighted* initial cost $\ell_{\Delta E_k}^I$, based directly on (26). Once again, we employ the ground contact and force constraints (12), (15), this time on the whole TO horizon. More importantly, we impose a joint impedance controller with

$$\tau_l - K_p \circ (q_l - \hat{q}_{12}) - K_d \circ (\dot{q}_l - 0) = 0 \quad (38)$$

where τ_l , q_l , \dot{q}_l refer to the actuated joint only, $K_p \in \mathbb{R}^2$ and $K_d \in \mathbb{R}^2$ are, respectively, stiffness and impedance gain vectors, which are to be optimized and \hat{q}_{12} is the reference configuration for the impedance controller, which is kept constant through the horizon and equal to the actuated joints of the landing configuration. The symbol \circ is used to indicate the element-wise product between the operands. Our final braking problem takes the form

$$\min_{W_b} L_b(W_b) \quad (39a)$$

subject to

$$\text{rigid body dynamics (1)} \quad (39b)$$

$$\text{foot tip contact (12), (15) up to } T_f \quad (39c)$$

$$\text{joint impedance control (38)} \quad (39d)$$

$$\text{torque limits (11)} \quad (39e)$$

$$\text{impact dynamics (22); } t = 0 \quad (39f)$$

$$\text{post-impact velocity: } v^+ = v; t = 0 \quad (39g)$$

$$\text{landing configuration: } q = \hat{q}; t = 0 \quad (39h)$$

$$\text{joint position and velocity ranges} \quad (39i)$$

where

$$W_b = [x_b, u_b, K_p, K_d, \hat{q}_{12}] \quad (40)$$

are our optimization variables and

$$L_b(W_b) = \ell_{\Delta E_k}^I + \int_0^{T_f} \ell_b(\tau) d\tau \quad (41)$$

is the employed cost function. The only parameter over which (39) depends is the pre-impact velocity v^- (through the impact dynamics constraint (22)) which, under the simplifications outlined in Sec. III-D, can be approximated with (30). Given the apex tip height h_{tip} provided by the solution of problem (21), we can compute a reasonable estimate for v^- as $[-\sqrt{2gh_{\text{tip}}}, 0, 0]$, where g is the acceleration of gravity.

IV. RESULTS

The TO optimization problems (21) and (39) are conveniently written employing the API provided by the Horizon tool [19] and solved on a 11th Gen Intel(R) Core(TM) i7-11800K CPU, using Ipopt and MA57 linear solver for step computations. The obtained optimal results are then tested both in simulation and on the real hardware. Our leg prototype is based on a 2 DOF leg prototype the joints of which are actuated by BLDC motor drives with 50:1 gear ratio. The power setup of the leg makes use of a 25Ah (100A peak discharge current), 51.2V nominal voltage battery unit and the energy flow from and to the battery is measured through a dedicated data acquisition system that monitors the power bus voltage and current levels.

A. Take-off generation

The optimal take-off trajectory is generated solving the TO problem (21) with the pipeline described in Sec. III-C: first a coarse problem made of 100 nodes is solved, then the obtained solution is re-sampled [19] at 1KHz (corresponding to 917 nodes for the optimal phases duration found solving (21)) and then refined with the same rate and number of nodes, so that a dynamically feasible trajectory is found. The solution of (21) is found in 79 iterations for the coarse initial problem and in 48 iterations for the refined one (see Fig. 11). Figure 9 shows part of the take-off TO solution and highlights how our formulation is able to exploit the performance of the hardware, while staying inside its limits.

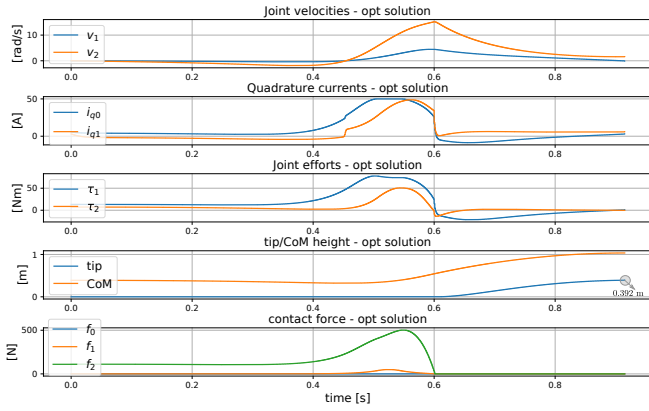


Fig. 9. Joint velocities, quadrature currents and joint efforts and jump height obtained from the refined take-off trajectory optimization problem. The limits for the quadrature current of 50 A (for safety lower than the maximum peak i_q of 62 A which the control boards can tolerate) are saturated. The peak torque limits of 120 Nm as well as the maximum link-side velocity of 20 rad/s, on the contrary, are not saturated.

B. Landing configuration and landing optimization

As detailed in Section III-F, once a solution of the problem (9) is available, it is possible to estimate the expected touchdown velocity. In this case, the final tip height is approximately 0.37m, for an estimated impact velocity of roughly 2.7m/s (computed as described at the end of Sec. III-F). Using this value as the input to (39), we solve (39) and obtain the optimal landing impedances and configuration

$$\begin{aligned} K_p^{\text{opt}} &= [10.65, 23.49] \text{ Nm/rad} \\ K_d^{\text{opt}} &= [6.92, 1.67] \text{ Nm/(rad/s)} \\ \hat{q}_{12} &= [1.34, 1.33] \text{ rad} \end{aligned} \quad (42)$$

after a total of 147 iterations (see Fig. 11). Looking at Fig. 5, the optimal \hat{q}_{12} is almost on the same curve of the configurations n.1 and n.2, with an impact ratio ρ of approximately $-3.84 \text{ N s}^2/\text{m}$, an impulse \hat{f} of 10.73 Ns and a total of 37.19 J of theoretical recovered energy. Figure 10 shows the power flow during braking computed from the final optimal solution (42) (top) and from a solution obtained without energy recovery maximization cost. Specifically, it is interesting to note how in the case of (42), the regenerative power trajectory follows a “critically damped” evolution.

C. Take-off execution

The obtained take-off trajectory is replayed on the real robot exploiting the XBot2 middleware [22] in a rt-safe control plugin running at 1KHz. employing a joint space impedance controller with high impedance set points. Specifically, the optimal trajectory is fed to the robot employing both position and velocity references. Figure 12 shows the results of the execution of a single jumping trajectory on the real robot (snapshots of the leg during the execution of the experiment are visible in Fig. 1). The comparison with the nominal optimal solution 9 reveals a good agreement in the peak currents and torques, thus confirming again the accuracy of our calibrated actuation model and the efficacy of the employed TO formulation.

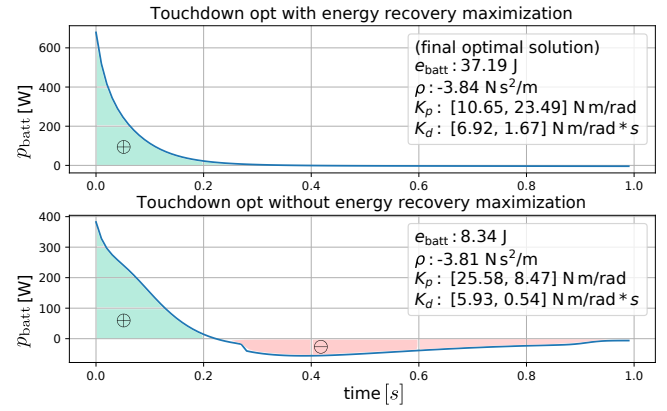


Fig. 10. Comparison of power flow towards the battery during braking, as computed by the landing optimization, without the energy recovery maximization cost (bottom) and with (top). Additionally, for each plot, also the regenerated energy e_{batt} , the optimized impedance gains K_p, K_d and the impact ratio ρ (32) are shown.

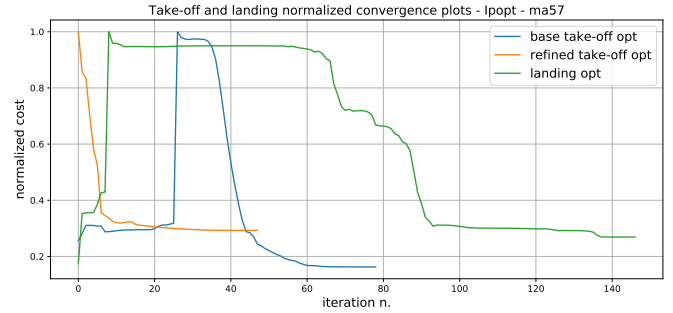


Fig. 11. Normalized convergence plots for the solution of the take-off and landing optimizations.

D. Energy regeneration results

In order to truly evaluate the correctness of the solution (42), we devised a cyclic jumping test. The optimal take-off trajectory is first replayed and, while the leg is still in the air, the impedance and position reference parameters of the controller are ramped to the optimal K_d, K_p and \hat{q}_{12} given by (42). The leg lands, we collect the estimated recovered energy, we ramp back up the impedance set points and then we repeat the same sequence up to the 15th jump in simulation and up to the 29th on the real hardware. Furthermore, we run again the full experiment with reasonable, but not optimized impedance set points, and compare the difference in the recovered energy. The results are summarized in Fig. 13, where we can appreciate how the optimized landing configuration and impedance combo is able to regenerate approximately 7 times more energy than the other ones in simulation and around 4 times more on the real hardware. To put these results in perspective, the drop-down tests shown in Fig. 7 and Fig. 6, were performed by letting the leg fall from an height of 0.5 m, which produced an impact velocity of approximately 3.16m/s, greater than the one obtained with the optimal jump. Even though the available pre-impact kinetic energy during the drop-down tests is 1.2 times higher, the optimal solution is still able to outperform non-optimized impedances and landing configurations in terms of energy recuperation.

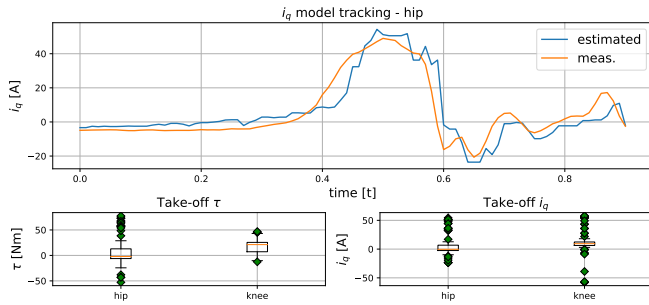


Fig. 12. Data from multiple executions of the optimal take-off trajectory on the real robot. On top, an further validation of the i_q model (shown for the hip joint and a single jump). At the bottom, the measured currents (right) and torques (left) during jumps, which are in line with the computed values shown in Fig. 9.

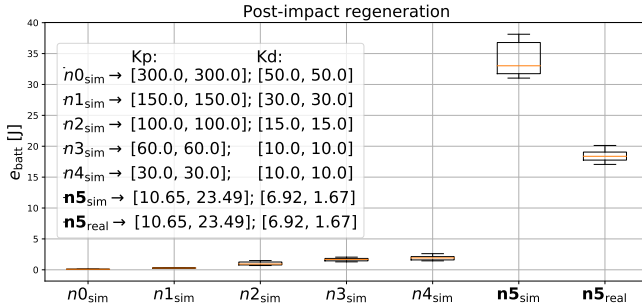


Fig. 13. Regenerated energy e_{batt} at touchdown after the execution of the optimal take-off trajectory, measured using 6 different impedance setpoints and landing configurations \hat{q}_{12} , including the optimal ones (shown in bold-face). In the non-optimal cases, the landing configuration is simply set to the one at the apex of the takeoff trajectory, i.e. $q_1 = 0.44$ rad, $q_2 = 0.66$ rad, with an associated impact ratio ρ (32) of -4.65 Ns²/m. The differences of 37.19 J (see Sec. IV-B) are attributable to the inevitable mismatches in both landing velocity and configuration which arise during the pre-landing flight phase. To avoid damages to the prototype, given the high number of jumps to be performed and the experimentally validated accuracy of (34), we decided to test the set points from $n0$ to $n4$ only in simulation.

V. CONCLUSIONS, CHALLENGES AND FUTURE WORK

In this work we presented a case study built around the generation and execution of agile jumping maneuvers on a quadruped leg prototype. We explored the peculiar challenges associated with jumping motions and proposed a multi-stage solution to the generation of agile jumping maneuvers made of two sequential TO problems for separately handling the thrust and the landing phases. The results of the optimizations, as well as our actuation and power models, were successfully validated both in simulation and on our quadruped leg prototype, showing the effectiveness of our approach in maximizing hardware performance and the usefulness of exploiting combined impact-energy considerations during landing phases.

One of the main limitations of our approach is its computational complexity, which currently does not allow for online re-planning. Future work will involve the extension of our pipeline and, specifically, of the combined energy recuperation - impact mitigation concept to a full-fledged legged robot, to more complex and online locomotion tasks, as well as to other phases, e.g. the take-off. This will require improvements of the current formulation and possibly the

use of more efficient solvers.

All the code employed to solve the optimizations, the data, as well as the scripts used to produce the plots are made publicly available at [23].

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon Europe Framework Programme under grant agreement No 101070596.

Furthermore, the authors would like to thank the ADVR Facility, Yifang Zhang and Davide Torielli for the crucial technical support.

REFERENCES

- [1] Hutter *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ IROS*, pp. 38–44, IEEE, 2016.
- [2] "Atlas gets a grip," YouTube, Feb. 23, 2023 [Online].
- [3] Nguyen *et al.*, "Optimized jumping on the mit cheetah 3 robot," in *2019 IEEE ICRA*, pp. 7448–7454, IEEE, 2019.
- [4] Chignoli *et al.*, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *2020 IEEE-RAS 20th ICHR (Humanoids)*, pp. 1–8, IEEE, 2021.
- [5] Roscia *et al.*, "Orientation control system: Enhancing aerial maneuvers for quadruped robots," *Sensors*, vol. 23, no. 3, p. 1234, 2023.
- [6] Katz *et al.*, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 IEEE ICRA*, pp. 6295–6301, IEEE, 2019.
- [7] Hawkes *et al.*, "Engineered jumpers overcome biological limits via work multiplication," *Nature*, vol. 604, no. 7907, pp. 657–661, 2022.
- [8] I. D. Walker, "Impact configurations and measures for kinematically redundant and multiple armed robot systems," *IEEE transactions on robotics and automation*, vol. 10, no. 5, pp. 670–683, 1994.
- [9] F. C. Anderson and M. G. Pandy, "Storage and utilization of elastic strain energy during jumping," *Journal of biomechanics*, vol. 26, no. 12, pp. 1413–1427, 1993.
- [10] Biewener *et al.*, "Elastic energy storage in the hopping of kangaroo rats (*dipodomys spectabilis*)," *Journal of Zoology*, vol. 195, no. 3, pp. 369–383, 1981.
- [11] Kashiri *et al.*, "Centauro: A hybrid locomotion and high power resilient manipulation platform," *IEEE RAL*, vol. 4, no. 2, pp. 1595–1602, 2019.
- [12] Yoong *et al.*, "Studies of regenerative braking in electric vehicle," in *2010 IEEE CSUDET*, pp. 40–45, IEEE, 2010.
- [13] Seok *et al.*, "Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot," *Ieee/asmе transactions on mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014.
- [14] Krause *et al.*, *Analysis of electric machinery and drive systems*, vol. 75. John Wiley & Sons, 2013.
- [15] L. Tien *et al.*, "Friction observer and compensation for control of robots with joint torque measurement," in *2008 IEEE/RSJ ICRIS*, pp. 3789–3795, 2008.
- [16] Ferrolho *et al.*, "Inverse dynamics vs. forward dynamics in direct transcription formulations for trajectory optimization," in *2021 IEEE ICRA*, pp. 12752–12758, IEEE, 2021.
- [17] Mastalli *et al.*, "Inverse-dynamics mpc via nullspace resolution," *arXiv preprint arXiv:2209.05375*, 2022.
- [18] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.
- [19] Ruscilli *et al.*, "Horizon: a trajectory optimization framework for robotic systems," *Frontiers in Robotics and AI*, vol. 9, 2022.
- [20] I. D. Walker, "The use of kinematic redundancy in reducing impact and contact effects in manipulation," in *Proceedings., IEEE ICRA*, pp. 434–439, IEEE, 1990.
- [21] Tassi *et al.*, "Impact planning and pre-configuration based on hierarchical quadratic programming," in *2022 IEEE ICRA*, pp. 1433–1439, IEEE, 2022.
- [22] A. Laurenzi *et al.*, "The xbot2 real-time middleware for robotics," *IEEE RAS*, vol. 163, p. 104379, 2023.
- [23] A. Patrizi, "HHCM-awesome-leg," https://github.com/ADVRHumanoids/awesome_leg/tree/humanoids23_submission, 2023. [Online; accessed 03-July-2023].