2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

CLIO: a Novel Robotic Solution for Exploration and Rescue Missions in Hostile Mountain Environments

Michele Focchi[1], Mohamed Bensaadallah[2], Marco Frego[3], Angelika Peer[3], Daniele Fontanelli[4], Andrea Del Prete[4], Luigi Palopoli[1]

***Abstract*— Rescue missions in mountain environments are hardly achievable by standard legged robots—because of the high slopes—or by flying robots—because of limited payload capacity. We present a concept for a rope-aided climbing robot which can negotiate up-to-vertical slopes and carry heavy payloads. The robot is attached to the mountain through a rope, and it is equipped with a leg to push against the mountain and initiate jumping maneuvers. Between jumps, a hoist is used to wind/unwind the rope to move vertically and affect the lateral motion. This simple (yet effective) two-fold actuation allows the system to achieve high safety and energy efficiency. Indeed, the rope prevents the robot from falling while compensating for most of its weight, drastically reducing the effort required by the leg actuator. We also present an optimal control strategy to generate point-to-point trajectories overcoming an obstacle. We achieve fast computation time (<1 s) thanks to the use of a custom simplified robot model. We validated the generated optimal movements in Gazebo simulations with a complete robot model with a $< 5\%$ error on a 16 m long jump, showing the effectiveness of the proposed approach, and confirming the interest of our concept. Finally, we performed a reachability analysis showing that the region of achievable targets is strongly affected by the friction properties of the foot-wall contact.**

I. Introduction

A climbing robot is a robot that moves on a vertical surface. The idea was first suggested in a seminal work by Nishi et al.[1] and has been revisited throughout for different application domains, including nuclear power plants [2], high-rise building cleaning [3], bridge maintenance [4], search and rescue missions [5]. However, there are still ways to climb walls not yet studied or reported [6].

The research work done so far on climbing robots is reviewed and classified in [7] and [8], respectively, according to the adhesion technique (electrostatic, magnetic, pneumatic, etc.) [9] used to attach the robot to the wall surface, and locomotion mechanism (i.e. wheel-Driven, tracked, legged, etc.) [10], utilized for climbing. The attachment and motion mechanisms are basically considered as the main problems when designing climbing robots. Additional requirements of equal importance come from the application and include (R1) safety, (R2) quickness in emergency situations, (R3) carrying payloads, and (R4) avoiding obstacles [11]. f

[1] The authors are with the Dipartimento di Ingegneria and Scienza dell'Informazione (DISI), University of Trento. Email: name.surname@unitn.it

[2] The author is with the Department of Electronics, University of Batna 2, Algeria. Email: m.bensaadallah@univ-batna2.dz

[3] The authors are with the Faculty of Science and Technology, Free University of Bozen-Bolzano. Email: name.surname@unibz.it

[4] The authors are with Dipartimento di Ingegneria Industriale (DII), University of Trento Email: name.surname@unitn.it. The publication was created with the co-financing of the European Union FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062 / 2021.

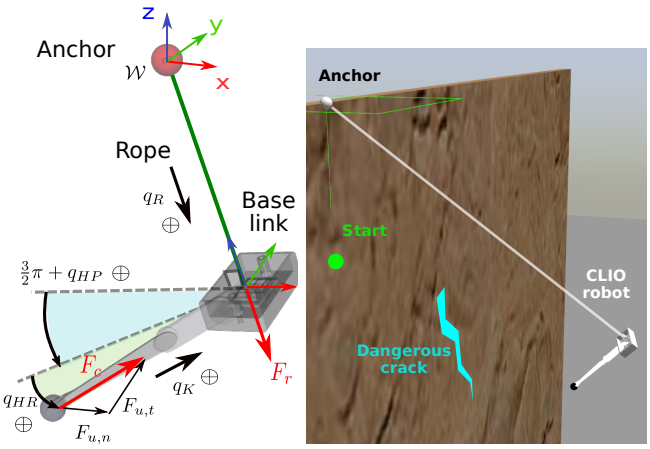

Fig. 1. (left) Kinematic model of the CLIO robot with standard definitions. The anchor frame is aligned with an inertial ($\mathcal{W}$) frame. (right) Realistic Gazebo simulation of a jump.

Human climbers inspired the use of a rope mainly for safety measures (R1) to support the weight of people during facade cleaning, firefighting and rescue missions [3]. In this paper, we combine the use of a rope with a leg, which can be retracted and extended very quickly. This allows the robot to jump away from the mountain wall, while the rope can be used to control its motion.

Additionally, as shown by Wang et al. [12] on dragline locomotion bio-inspired from spiders, the aid of rope can dramatically increase the locomotion speed (R2) by a winding and releasing mechanism, therefore being a preferable solution in applications that require a prompt intervention, such as Search & Rescue missions. Furthermore, ropes can support much heavier payloads (R3) than robots relying on sticky/adhesive approaches [13], [14] because of the limited tangential component of the adhesive and leg actuation. However, having the robot attached to a rope poses some challenges. First, the robot is under-actuated because it cannot *fully* control the position of its center of mass when not in contact with the wall. Second, the rope represents a *unilateral* constraint (i.e. it can only pull and not push), which further complicates the already hybrid dynamics and the low control authority of this class of robots. Finally, rather than slowly taking steps as a walking mechanism, the robot can take one or multiple jumps like the rapid jumping Salto-1P shown in [15] to reach the desired locations, and overcoming obstacles. Some early work on planning jumping trajectories for a rope-aided robot has been performed in [16], but no physical validation was reported.

To summarize, the key features of jumping with a rope are related to the locomotion speed and the ability to address up to *vertical* inclinations. Moreover, the resulting motion of the robot (and so the possibility to reach the target) depends on: 1) the impulse exerted on the wall at lift-off, 2) the winding/unwinding of the rope. Therefore, a planning strategy for this kind of motions should take into account both these factors, besides the under-actuation, the constraints posed by the rope, the actuator limits and the contact interaction (i.e. friction).

To tackle these issues, numerical optimization comes as an attractive solution for these planning problems [17], [18], since it allows to minimize some optimality criteria while ensuring that the physical constraints are satisfied along the planning horizon. In this framework, different goals can also be pursued, such as minimizing energy consumption or reaching the target in minimum time.

In this work, we present a robotic platform called CLIO (CLImbing rObot) that is able to reach desired targets on a vertical (or slanted) wall with different frictional properties. We also propose a planning approach based on numerical optimization, to solve the jumping problem employing a simplified model of the dynamics. Hence, the contributions of the paper can be summarised in what follows:

- A conceptual design of a jumping robot platform CLIO;
- An optimal control formulation to generate a jump motion to reach a desired target while overcoming an obstacle, based on a simplified model of the system, which results in reduced computation time;
- Simulation experiments to validate the effectiveness of the proposed approach, both using the simplified model and in a more realistic (Gazebo) simulation, considering the full dynamics of the robot.

As an additional result, we propose a reachability analysis that shows that the region of achievable targets is limited by the friction coefficient at the lift-off location.

The paper is organized as follows: Section II gives an overview of the robot platform and derives a simplified model of it. Section III describes the optimization problem to plan jump trajectories based on the simplified model. Simulation results are presented in Section IV. Finally, we draw the conclusions in Section V.

II. Robot Description

A. Full robot description

This section presents the kinematic model of the CLIO robot (see Fig. 1(left)). We model the robot as a serial kinematic chain with $n = 9$ Degrees of Freedom (DoFs) represented by the configuration vector $\mathbf{q} \in \mathbb{R}^n$. We model the attachment between the anchor point (a fixed link) and the rope with 2 *passive* (rotational) joints. The rope is modeled as an *actuated* prismatic joint (q_R), followed by 3 *passive* rotational joints to model the connection between the rope and the *base link* of the robot[1].

[1]It would be equivalent to allocate 3 joints at the anchor point and 2 at the base. To avoid redundancy in the representation, it is necessary to have only one passive joint along the rope axis.

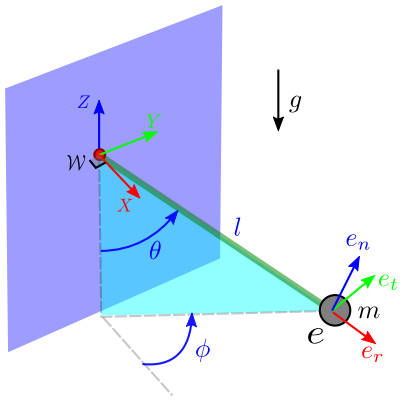

Fig. 2. Logical scheme of the simplified model

The propulsion mechanism that allows the robot to jump is a 3-DoFs robotic leg with a point-like foot that enables to exert a *pure* Cartesian force (no moment) on the wall. The leg has two *subsequent* rotational joints, called hip pitch (q_{HP} about the Y axis (green)) and hip roll (q_{HR} about the X axis (red)). These joints are useful to align the leg to the thrusting impulse. This enables to avoid *centroidal* moments that would pivot the robot around the rope axis. Finally, a prismatic knee (q_K) joint is used to generate the *thrusting* impulse. With the actual design, the robot will not be able to stabilize itself on the wall. However, the design of a landing and stabilization mechanism is out of the scope of this work, where we focus on the jump.

The dynamics equation can be written as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{0}_{5\times 1} \\ \boldsymbol{\tau}_a \end{bmatrix} + \mathbf{J}^T \mathbf{F}_c \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the inertia matrix; $\mathbf{h} \in \mathbb{R}^n$ represents the bias terms (Centrifugal, Coriolis and Gravity); and $\mathbf{J} \in \mathbb{R}^{3\times n}$ is the Jacobian relative to the contact point mapping the contact force $\mathbf{F}_c \in \mathbb{R}^3$. Unless specified, all vectors are expressed in an inertial $\mathcal{W}$ frame (attached to the anchor). The under-actuation is captured by the vector $\mathbf{0}_{5\times 1}$, while the efforts of the actuated joints are grouped into the vector $\boldsymbol{\tau}_a = \begin{bmatrix} \boldsymbol{\tau}_R & \boldsymbol{\tau}_{\text{leg}} \end{bmatrix}^T \in \mathbb{R}^4$.

B. Simplified Model

Dealing with the full dynamics of the robot can be time consuming, due to the high number of states and constraints involved. Therefore, in this section, we derive a lower dimensional model (3 DoFs) that captures the dominant dynamics of the system by making the following assumptions: 1) the mass is entirely concentrated in the body attached to the rope, 2) we can unwind and rewind the rope that is assumed to be inextensible, rigid, and remains completely elongated (i.e., it cannot bend), 3) the rewinder mechanism can pull the rope when winding, but it cannot push (i.e., it can only act as a brake during the unwinding phase). A geometric sketch of the system is shown in Fig. 2, where the pitch angle is θ, the yaw angle is ϕ, and l is the length of the rope.

Our input variables to the system are: 1) the rewinder pull action or braking action $\mathbf{F}_r$ oriented along the rope, 2) an

impulsive pushing force $\mathbf{F}_u$ that the robot can generate when it is attached to the mountain wall.

Let e be a frame attached to the point mass m, and let $\mathcal{W}$ be the world frame attached to the anchor point of the pendulum. If we consider the homogeneous transformation from e to $\mathcal{W}$, we can write:

$$\mathbf{T}_e^{\mathcal{W}} = \begin{bmatrix} \mathbf{R}_z(\phi) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_y(\frac{\pi}{2}-\theta) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} & & l \\ \mathbf{I}_{3\times3} & & 0 \\ & & 0 \\ \mathbf{0}_{1\times3} & & 1 \end{bmatrix}$$

Simplifying $\mathbf{T}_e^{\mathcal{W}}$, gives

$$\mathbf{T}_e^{\mathcal{W}} = \begin{bmatrix} c_\phi s_\theta & -s_\phi & c_\phi c_\theta & l c_\phi s_\theta \\ s_\phi s_\theta & c_\phi & c_\theta s_\phi & l s_\phi s_\theta \\ -c_\theta & 0 & s_\theta & -l c_\theta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where c_x is a shorthand for $\cos x$, and s_x is a shorthand for $\sin x$. The position $\mathbf{p}$ of the mass can be extracted from the transformation matrix $\mathbf{T}_{\mathcal{B}}^{\mathcal{W}}$ as:

$$\mathbf{p} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} l s_\theta c_\phi \\ l s_\theta s_\phi \\ -l c_\theta \end{bmatrix} \tag{2}$$

while the axes $\mathbf{e}_r$, $\mathbf{e}_t$, $\mathbf{e}_n$ are the columns of the top-left sub-matrix of $\mathbf{T}_e^{\mathcal{W}}$, respectively. From equation (2), the velocities along the Cartesian axes are:

$$\begin{aligned} \dot{X} &= l c_\theta c_\phi \dot{\theta} - l s_\theta s_\phi \dot{\phi} + s_\theta c_\phi \dot{l} \\ \dot{Y} &= l c_\theta s_\phi \dot{\theta} + l s_\theta c_\phi \dot{\phi} + s_\theta s_\phi \dot{l} \\ \dot{Z} &= l s_\theta \dot{\theta} - c_\theta \dot{l} \end{aligned}$$

Next, the velocity of the mass squared:

$$v^2 = \dot{X}^2 + \dot{Y}^2 + \dot{Z}^2 = l^2\dot{\theta}^2 + l^2 s_\theta^2 \dot{\phi}^2 + \dot{l}^2$$

Thus, the kinetic energy K and potential energy V:

$$\begin{aligned} K &= \frac{m}{2} v^2 = \frac{m}{2} l^2 \left(\dot{\theta}^2 + s_\theta^2 \dot{\phi}^2 \right) + \frac{m}{2} \dot{l}^2 \\ V &= mgz = -mgl c_\theta \end{aligned}$$

which leads to the Lagrangian function described as:

$$L = K - V = \frac{m}{2} l^2 \left(\dot{\theta}^2 + s_\theta^2 \dot{\phi}^2 \right) + \frac{m}{2} \dot{l}^2 + mgl c_\theta \tag{3}$$

The dynamics of the system will be obtained using the Euler-Lagrange equation:

$$\begin{aligned} &\ddot{\theta} + \frac{2}{l}\dot{\theta}\dot{l} - c_\theta s_\theta \dot{\phi}^2 + \frac{g}{l} s_\theta = \frac{1}{ml} F_{u,n} \\ &\ddot{\phi} + 2\frac{c_\theta}{s_\theta}\dot{\theta}\dot{\phi} + \frac{2}{l}\dot{\phi}\dot{l} = \frac{1}{ml s_\theta} F_{u,t} \\ &\ddot{l} - l\dot{\theta}^2 - l s_\theta^2 \dot{\phi}^2 - g c_\theta = \frac{1}{m} F_r. \end{aligned} \tag{4}$$

Clearly, in this derivation of the model, we have heavily relied on the point-mass nature of the body. A possible issue could be the rotational dynamics of the body around the rope axis, which could waste energy, generating unnecessary motions and impede the landing phase. This issue will be part of our future work. However, as discussed next, under reasonable assumptions (e.g., thrusting force oriented in the direction of the center of mass), the results based on the simplified model can be applied to a realistic system with a good approximation.

C. Robot motion: problems and solution overview

The problem of moving CLIO between two given configurations is not easy. First of all, even the simplified dynamics in (4) is highly nonlinear. Second, the configurations between which the robot moves are usually distant, and therefore, using a linearization of the dynamics equations 4 around a specific operating point would lead to inaccurate results. Third, one of our actuators, the thrusting force $\mathbf{F}_u$, has an impulsive nature and operates at discrete time instants. Therefore, it is not possible to use it in any feedback control scheme. Finally, the tangential component $F_{u,t}$ is generated by using friction. Therefore, the two components $F_{u,n}$ and $F_{u,t}$ are coupled by the nonlinear friction cone constraint. Because the friction cone depends on the specific point or area where the robot is pointing its leg, this mechanism is not totally reliable (i.e., there can be significant deviations between generated and desired values of $F_{u,t}$).

In view of this complexity, we propose an approach based on two steps: 1. a motion strategy is planned prior to starting the motion, 2. after take off, a feedback motion controller operates on $\mathbf{F}_r$ to compensate for small deviations and secure that the robot lands close to the expected position. In this paper we will mainly focus on step 1.

III. Motion Planning

The motion plan for the robot is decided by solving an optimal control problem *à la* Pontryagin. Generally speaking, we can set up the problem in the following way:

$$\begin{aligned} &J = \min_{\mathbf{u}(t)} M(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) + \int_{t_0}^{t_f} L\left(\mathbf{x}(t), \mathbf{u}(t)\right) dt \\ &\text{subject to:} \\ &\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ &\mathbf{u} \in \mathcal{U}, \\ &\mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0, \\ &\mathbf{B}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0, \end{aligned} \tag{5}$$

where $\mathbf{x}$ is the vector of state variables, $\mathbf{u}$ is the control input constrained in the convex set $\mathcal{U}$, f is the dynamic equation of the system, $\mathbf{h}$ is the constraint on the state and control variables, $\mathbf{B}$ are the boundary conditions, t_0 and t_f are the initial and final time (that is an optimization variable), which can be part of the optimization. The OCP (5) is in the standard form with Mayer term M and Lagrange term L that model initial/terminal costs and the running cost, respectively.

Dynamical System. The state $\mathbf{x}$ of our problem is $\mathbf{x} = \left[\theta,\, \phi,\, l,\, \dot{\theta},\, \dot{\phi},\, \dot{l}\right]^T$, while the control variables are given by

$\mathbf{u} = [\mathbf{F}_u,\, \mathbf{F}_t]^T$. The dynamic equation is derived from (4):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ -\frac{2}{x_3}x_4x_6 + c_{x_1}s_{x_1}x_5^2 - \frac{g}{x_3}s_{x_1} + \frac{1}{mx_3}F_{u,n} \\ -2\frac{c_{x_1}}{s_{x_1}}x_4x_5 - \frac{2}{x_3}x_5x_6 + \frac{1}{mx_3s_{x_1}}F_{u,t} \\ x_3x_4^2 + x_3s_{x_1}^2x_5^2 + gc_{x_1} + \frac{1}{m}F_r. \end{bmatrix} \tag{6}$$

Boundary Conditions. The terminal constraints are usually expressed in Cartesian space $[X,\, Y,\, Z,\, \dot{X}, \dot{Y}, \dot{Z}]^T$ and they can be expressed as a function of the state variables by means of (2). For instance, for X, Y, Z, we have:

$$\begin{aligned} x_1 &= \text{atan2}\left(-Z, \sqrt{X^2+Y^2}\right), \\ x_2 &= \text{atan2}(Y, X), \\ x_3 &= \sqrt{X^2+Y^2+Z^2}, \end{aligned} \tag{7}$$

State Constraints. State constraints are related to regions of the operation space that are not accessible. For instance, an irregular form of the walls could generate obstacles that the robot has to overcome in order to reach its final destination. Generally speaking, the inaccessible area is modelled as a region whose boundary is a surface. We assume that this surface can be expressed by a differential 2D manifold expressed by $f(\mathbf{x}) = 0$. Therefore the admissible region is generally given by $f(\mathbf{x}) \geq 0$ and could be non-convex. Another constraint is added to prevent collision with the wall (i.e. $X > 0$) and to ensure the robot elevation does not exceed the anchor level (i.e. $Z < 0$).

Actuation Constraints. The system actuators are given by the two forces $\mathbf{F}_r$ and $\mathbf{F}_u$. The $\mathbf{F}_r$ force acts along the direction of the rope ($\mathbf{e}_R$ axis). Since the rope cannot push the robot (assumption 3, the rewinder can only pull the wire), F_r can only be used to retract the robot or to slow down its descent (under the action of gravity). Moreover, the force is bounded by the limits of the actuators (e.g. a hoist). Therefore, the constraint on F_r is:

$$-F_r^{\max} \leq F_r \leq 0.$$

As regards $\mathbf{F}_u$, since it is meant to act for a very short duration, reaching high peaks, we model it through a Dirac impulse: $\mathbf{F}_u(t) = \mathbf{F}_u\delta(t)$, where $\mathbf{F}_u$ is a vector with the two components $F_{u,t}$ and $F_{u,n}$. The Dirac delta is a generalized function and cannot be handled by the optimal control frameworks, therefore, a smooth approximation is needed. A natural approximation is a Gaussian function, that is $\delta(t - t_0) \approx \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(t-t_0)^2}{2\sigma^2}}$, [19]. The duration T_{th} of the impulse is roughly given by 6σ. Given the nature of our actuation mechanism, a realistic choice is to have a duration in the order of tens of milliseconds, which consequently determines a remarkable intensity of the impulse. The maximum norm of $\mathbf{F}_u$ is upper-bounded by the actuation limits:

$$\sqrt{F_{u,n}^2 + F_{u,t}^2} \leq F_u^{\max} \quad .$$

Additionally, the tangential component $F_{u,t}$ is generated by the friction with the mountain wall. Hence, $F_{u,n}$ and $F_{u,t}$ are constrained by the following relation (friction cone):

$$|F_{u,t}| \leq \mu F_{u,n}$$

where μ is the *friction coefficient*, a constant depending on the nature of the terrain.

Objective Function. The cost is composed of three terms. First, we minimize the time t_f to complete the mission (assuming without loss of generality that $t_0 = 0$). Second, we minimize the final kinetic energy $K(t_f)$ because when the robot lands, it has to dissipate the energy through a damper (we are not considering to reuse the accumulated energy to re-bounce after landing). Third, we minimize the work performed by the winding mechanism. Thus, the cost is:

$$J = w_t t_f + w_E K(t_f) + w_{F_r}\int_0^{t_f} (F_r \cdot x_6(t))^2 dt,$$
$$K(t) = \frac{m}{2}x_3(t)^2\left(x_4(t)^2 + s_{x_1(t)}^2 + x_5(t)^2\right) + \frac{m}{2}x_6(t)^2,$$

with w_t, w_E and w_{F_r} being the weights associated to the three cost components. We should observe that, in case of a difficult convergence, it is possible to "soften" some of the constraints. For instance, for the terminal constraint, it is possible to introduce a slack variable Δ and have a constraint $\|\mathbf{p}(t_f) - \mathbf{p}_f\|^2 \leq \Delta$, where the slack Δ can be either a maximum tolerance set by the user or become part of the cost function. Another possibility is to add the relaxed constraint as a penalty in the objective function.

Initial guess. To speed-up convergence, we initialized the optimization with a reasonable initial guess. Because time t_f is an optimization parameter, we compute the time constant for the system linearized around the initial state. The linearized system becomes also a good approximation when the jump length is small with respect to the rope elongation.

IV. Simulation Results

This section presents simulation results that show how the control inputs obtained running the optimization of Section III, are able to bring the robot to a desired target. We used the OCP solver PINS [20], [21], [22] which relies on the indirect approach based on the Pontryagin Maximum Principle. The code to replicate the results is freely available at [2].

In a first experiment, we run the optimization to perform a 12 m long jump starting from an initial position $\mathbf{p}_0 = \begin{bmatrix}0.24 & 0 & -8\end{bmatrix}^T$ up to a target at $\mathbf{p}_f = \begin{bmatrix}3 & 3 & -20\end{bmatrix}^T [m]$. As an additional constraint, during the jump, the robot has to avoid a rock pillar attached on the wall, that we model as a 20 m high cone with a base of radius 2.5 m. We validate the results of the optimization for both the simplified model (in a Matlab environment) and the full 3D model. For this case, we built a Gazebo simulator based on the URDF description [23] of the robot (see Fig. 1(right)). Table I reports the used physical parameters, together with some optimization settings. For the Matlab simulation, we simply integrate (4)

[2] https://github.com/mfocchi/climbing_robots.git

TABLE I

SIMULATIONS PARAMETERS

Name	Symbol	Value
Robot mass	m	5
Max. impulse [N]	$F_u^{\max}$	1000
Max. retraction force [N]	$F_r^{\max}$	200
Friction coeff.	μ	0.7
Thrust impulse duration [s]	T_{th}	0.025
Discretization steps	N	500

with an ode45 Runge-Kutta variable-step scheme. For the Gazebo simulation, we assume to apply the push impulse as a force F_c at the contact point. Therefore we employ the leg dynamics to define the mapping between the contact force F_c and the torques τ_{leg} at the leg joints:

$$\boldsymbol{\tau}_{\text{leg}}^d = \begin{bmatrix} \tau_{HP} \\ \tau_{HR} \\ \tau_K \end{bmatrix} = \mathbf{h}_{\text{leg}} - \mathbf{J}_{\text{leg}}^T \underbrace{\left(R_e^{\mathcal{W}} \begin{bmatrix} F_{u,n} \\ F_{u,t} \\ 0 \end{bmatrix} \right)}_{\mathbf{F}_c} \tag{8}$$

Where $\mathbf{J}_{\text{leg}} \in \mathbb{R}^{3\times 3}$ is the sub-matrix of the Jacobian $\mathbf{J}$ relative to the leg joints, and $\mathbf{h}_{\text{leg}} \in \mathbb{R}^3$ represents the bias terms. We do not generate any impulse along the rope direction ($\mathbf{e}_R$ axis) in order to avoid to accidentally create any slack on the rope. Additionally, to avoid angular motions, we implement virtual dampers to keep the passive joints at the base in a fixed position, since the optimization is performed considering the simplified model that neglects the angular dynamics.

We set the initial configuration to $\mathbf{q}_0 = \begin{bmatrix}\text{atan2}(r_{\text{leg}}, l_0) & 0 & 0 & l_0 & 0 & 0 & 0 & -1.57 & 0 & 0\end{bmatrix}^T$ where l_0 is the initial rope length and $r_{leg} = 0.32\ m$ is the leg length at the startup configuration. At q_0, the foot is meant to touch the wall in $\mathbf{p}_0$, the starting point of the optimized trajectory. A state machine coordinates the 3 phases of the jump: leg orientation, thrusting and flying. In the leg orientation phase, the hip roll joint set-point q_{HR}^d is commanded to have the leg aligned with the impulse direction, $q_{HR}^d = \text{atan2}(\max(F_{u,t}), \max(F_{u,n}))$. Note that the hip-pitch joint set-point is $q_{0,HP}^d = -1.57$ in order to have the leg lying on the X-Y plane of the base frame. A low level PD controller runs in parallel with the feed-forward actions $\boldsymbol{\tau}_{\text{leg}}^d$ to drive the joints. During the *thrusting* phase, the force $\mathbf{F}_c$ is generated at the contact applying $\boldsymbol{\tau}_{\text{leg}}^d$ for the thrust duration $T_{th} = 0.025s$, while the PD gains are switched off to avoid conflicts. Then it follows the flying phase, where the rope winding joint is actuated with the optimized force pattern $\tau_R = F_r(t)$ for the whole jump duration t_f. The computation time for the optimization and the integration error at the target $\|e_f\|$ are linearly linked to the number of discretization points N. Table II reports the integration error normalized for the jump length $\|\mathbf{e}_f\|/(l_f - l_0)$ for different numbers of discretization points. The Table shows that a good trade-off can be found between accuracy and computation time that enables fast computation. Fig. 3 reports the results for $N = 500$. Simulating the simplified model, the matching is almost perfect (apart from integration drifts), while in the full-model Gazebo simulation, the norm of the final error is around 0.75 m for a 16 m jump ($< 5\%$). This is due to a number of reasons: 1) the control inputs are applied in open-loop, therefore any uncertainty causes an error in the lift-off momentum that can cause drift during the jump, 2) the impulse is applied at the foot and not at the Center of Mass (CoM), 3) the approximation due to the simplified model with respect to the full one used in the Gazebo simulation. However, a 0.6% tracking error is in a range that can be efficiently coped with a controller, and shows that the simplified model is a good approximation for the real system.

TABLE II

RESULTS OF THE NUMERICAL OCP FOR DIFFERENT DISCRETISATIONS

N	Comp. time [s]	$\|\mathbf{e}_f\|$ [m]	$\|\mathbf{e}_f\|/(l_f - l_0)$ [%]
250	0.850	0.0937	0.76
500	1.472	0.0525	0.42
1000	2.648	0.0238	0.19
2000	5.063	0.0180	0.14

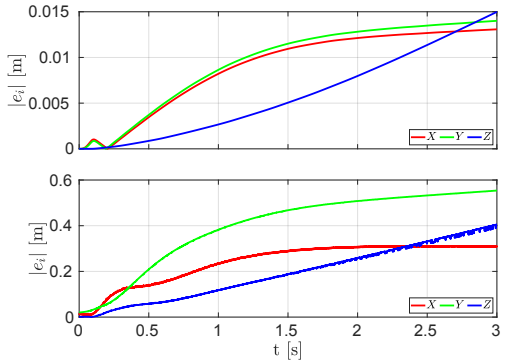

Fig. 3. Simulation. Validation of the optimization results. Absolute value of the validation error between the CoM trajectory computed by the optimization and the simulated trajectory with (upper plot) Matlab and (lower plot) Gazebo, respectively.

Additionally, we run the optimization to jump on 3 different targets on the rock pillar (see Fig. 4), starting from a point on the wall $\mathbf{p}_0 = \begin{bmatrix}0.24, 0, -8\end{bmatrix}$ (Experiment 1,2,3). To demonstrate that the approach is valid also for jumps from a *non vertical* (i.e. slanted) surface, Experiment 4 is a jump from a location $\mathbf{p}_0 = \begin{bmatrix}0.63 & 2.35 & -7.5\end{bmatrix}$ that is already on the pillar, which has an inclination of around 80 deg. Each optimization provided the max values of the initial impulses ($F_{u,n}$, $F_{u,t}$)[3], the pattern of the winding force F_r and the jump duration T_f. For the first 3 targets, we also plot the friction boundaries in the tangential direction at the lift-off point (red shaded area).

The results are reported in Table III and in the accompanying video[4]. As expected, the final kinetic energy (that will be lost at the impact) is higher for longer jumps, the jump duration, instead, is higher for the jumps that involve a bigger lateral motion. By inspecting the red trajectories in

[3]The value of these forces are related to the selected impulse duration T_{th} [s], they can be strongly reduced by taking longer durations (i.e. in accordance to the actuator response time).

[4]Video link: https://youtu.be/RZUfaaDkoR0

TABLE III

MULTIPLE TARGETS SIMULATIONS RESULTS

Exp.	$\mathbf{p}_f$ [m]	T_f [s]	$F_{u,n}$ [N]	$F_{u,t}$ [N]	K_f [J]
1	$[2.60, 2.73, -20.46]$	3.12	306	214	578
2	$[2.50, 0.05, -20.46]$	1.62	236	5.11	596
3	$[0.55, -0.8, -17.76]$	3.18	217	-151	422
4	$[1.73, 3.8, -20.46]$	1.69	168	-64	619

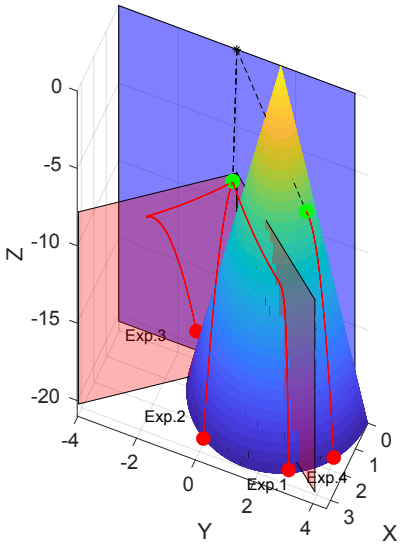

Fig. 4. Simulation. The black star is the anchor point, the green dot is the starting point $\mathbf{p}_0$ the red dot the target location $\mathbf{p}_f$. The blue shaded area is the rock face, while the red shaded area represents the friction boundaries ($\mu = 0.7$) in the tangential direction. The cone represents an obstacle (rock pillar) to be avoided.

Fig. 3, one can see that for the targets close to the friction boundaries, to "steer" the trajectory more laterally, the optimizer dictates an *initial* retraction of the rope. This strategy is meaningful, because lateral pushes are limited due to the limits on the tangential force posed by friction, therefore the only resort is to vary the time constant of the "pendulum" (i.e. reducing the inertia moment about the anchor point) by initially winding the rope. This decreases the deceleration on the ϕ variable due to the gravity component. Eventually, the rope is let to passively unwind (under the action of gravity) to attain the target. To visualize this fact and better assess the capabilities of the system, we perform a reachability analysis, plotting a top view (X-Y plane) of the region of reachable jump targets (see Fig. 5) lifting-off from the same point $\mathbf{p}_0$. We run a number of optimizations for a grid of targets below $\mathbf{p}_0$, in the positive Y half-space (the solutions are symmetric in the negative one). We plot a marker for the targets where the optimization convergence is successful to state that they are *reachable*. We also associate a color code to the energy consumed by the winding mechanism for each target. The friction boundary is marked as a red line. We notice that there are a few reachable targets out of the area limited by the friction cone and that they are associated with a very high consumption in terms of energy from the winding mechanism, because they involve a big initial retraction, corroborating the insights got in Fig. 4. On

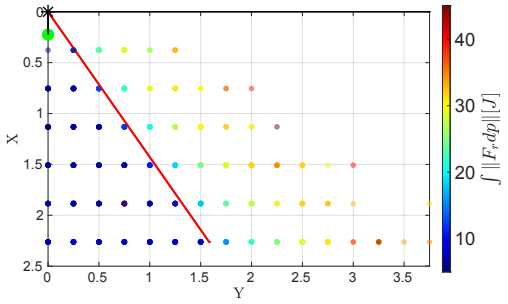

Fig. 5. Simulation. Top view (X-Y plane) of the region of reachable targets for a friction coefficient $\mu = 0.7$. The black star is the anchor point, the big green dot is the common lift-off point $\mathbf{p}_0$. The color bar indicates the energy consumed by the winding mechanism to reach the target.

the other hand, with a fixed rope, lateral jumps would be very limited. Indeed, the fact of performing small lateral jumps to move laterally, is also what experienced climbers do to realize a big pendulum swing on a rock face [24].

V. CONCLUSIONS

We presented a robotic solution for exploration and rescue missions in mountain environments such as canyons, lunar craters, etc. It could be employed in civil applications in scaling maintenance activity, e.g., to detach from the mountain walls dangerous boulders, needed to mitigate the hydro-geological risk. This would create additional market opportunities for a possible business. Our platform combines the use of a rope of adjustable length with a jumping mechanism. We validated the result of the optimization carried out using a simplified model with a fully-detailed model simulation in Gazebo, obtaining 5% discrepancy on a 16 m jump. The main limitations of the actual work are the limited *reachability* with the single anchor arrangement and the lack of descriptiveness (in terms of angular dynamics) of the simplified model. This prevents to devise planning strategies that optimize also the orientation of the robot at the landing. To address the first issue, we plan to investigate a solution with two ropes with more than one anchor point to increase the reachable region on the wall; while, for the second issue, we plan to release the point-mass approximation and considering the robot as a rigid body with non-trivial mass geometry. In this case, the thrust action has to be combined with a way to control the rotation of the body and in order to ensure the proper alignment of the leg to guarantee a safe landing. Last, we are working on the design of a landing controller to dissipate the excess of kinetic energy at landing, avoiding rebounces. All these steps are preparatory to the development of a working prototype. Many important future directions have been opened by this research. Whilst avoiding obstacles is a well-known problem for motion planning in horizontal terrain, the jump motion pattern determined by the ropes makes the motion-planning problem non-standard, calling for new approaches. Much work has to be done also in the area of motion control in order to ensure that the planned trajectory is closely tracked during the flight.

REFERENCES

[1] A. Nishi, Y. Wakasugi, and K. Watanabe, "Design of a robot capable of moving on a vertical wall," *Adv. Robot.*, vol. 1, no. 1, pp. 35–45, 1986.

[2] L. Briones, P. Bustamante, and M. A. Serna, "Wall-climbing robot for inspection in nuclear power plants." Publ by IEEE, 1994, pp. 1409–1414.

[3] S. Nansai and R. E. Mohan, "A survey of wall climbing robots: Recent advances and challenges," *Robotics*, vol. 5, 9 2016.

[4] R. Wang and Y. Kawamura, "Development of climbing robot for steel bridge inspection," *Industrial Robot*, vol. 43, pp. 429–447, 2016.

[5] M. Eich, F. Grimminger, F. Kirchner, and D. Bremen, "A versatile stair-climbing robot for search and rescue applications," 2008, pp. 35–40.

[6] M. Tavakoli and C. Viegas, *Bio-inspired climbing robots*. Elsevier Ltd., 2015. [Online]. Available: http://dx.doi.org/10.1016/B978-0-08-100249-0.00014-8

[7] Y. Fang, S. Wang, Q. Bi, D. Cui, and C. Yan, "Design and technical development of wall-climbing robots: A review," *Journal of Bionic Engineering*, 7 2022.

[8] A. Hajeer, L. Chen, and E. Hu, "Review of classification for wall climbing robots for industrial inspection applications," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1421–1426.

[9] D. Longo and G. Muscato, *Adhesion Techniques for Climbing Robots: State Of The Art and Experimental Considerations*. World Scientific, 2008. [Online]. Available: www.worldscientific.com

[10] B. Chu, K. Jung, C. S. Han, and D. Hong, "A survey of climbing robots: Locomotion and adhesion," *International Journal of Precision Engineering and Manufacturing*, vol. 11, pp. 633–647, 8 2010.

[11] D. Schmidt and K. Berns, "Climbing robots for maintenance and inspections of vertical structures - a survey of design aspects and technologies," *Robotics and Autonomous Systems*, vol. 61, pp. 1288–1305, 12 2013.

[12] L. Wang, U. Culha, and F. Iida, "A dragline-forming mobile robot inspired by spiders," *Bioinspiration and Biomimetics*, vol. 9, 3 2014.

[13] S. Kim, M. Spenko, S. Trujillo, B. Heyneman, D. Santos, and M. R. Cutkoskly, "Smooth vertical surface climbing with directional adhesion," *IEEE Transactions on Robotics*, vol. 24, pp. 65–74, 2 2008.

[14] D. K. Riskin and P. A. Racey, *How do sucker-footed bats hold on, and why do they roost head-up?*, 2009. [Online]. Available: https://academic.oup.com/biolinnean/article/99/2/233/2448125

[15] D. W. Haldane, J. K. Yim, and R. S. Fearing, "Repetitive extreme-acceleration (14-g) spatial jumping with salto-1p," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 3345–3351, 2017.

[16] E. M. Hoffman, M. Parigi Polverini, A. Laurenzi, and N. G. Tsagarakis, "Modeling and optimal control for rope-assisted rappelling maneuvers," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9826–9832.

[17] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the MIT cheetah 3 robot," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2019-May, pp. 7448–7454, 2019.

[18] Y. Ding, C. Li, and H. W. Park, "Kinodynamic motion planning for multi-legged robot jumping via mixed-integer convex program," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 3998–4005, 2020.

[19] A. Saichev and W. Woyczynski, *Distributions in the Physical and Engineering Sciences: Distributional and Fractal Calculus, Integral Transforms and Wavelets*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 1996.

[20] N. D. Bianco, E. Bertolazzi, F. Biral, and M. Massaro, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Vehicle System Dynamics*, vol. 57, no. 5, pp. 665–696, 2019. [Online]. Available: https://doi.org/10.1080/00423114.2018.1480048

[21] H. A., M. Gerdts, and E. Bertolazzi, "Structure exploitation in an interior-point method for fully discretized, state constrained optimal control problems," *Vietnam J. Math*, vol. 46, pp. 1089–1113, 2018.

[22] F. Biral, E. Bertolazzi, and P. Bosetti, "Notes on numerical methods for solving optimal control problems," *IEEJ Journal of Industry Applications*, vol. 5, no. 2, pp. 154–166, 2016.

[23] "URDF Package Summary." [Online]. Available: http://wiki.ros.org/urdf

[24] "Youbube - Yosemite King Swing - Nose route." [Online]. Available: https://www.youtube.com/watch?v=by8fCPW_Sas