

Modeling and Optimal Control for Rope-Assisted Rappelling Maneuvers

Enrico Mingo Hoffman, Matteo Parigi Polverini, Arturo Laurenzi, and Nikos G. Tsagarakis

Abstract—Envisioning the employment of rope-assisted humanoid robots to reduce human intervention for operations in the heights, this preliminary work addresses the modeling and motion planning problems for a rope-assisted bipedal robot. The mathematical features of this system outnumber the ones of typical humanoid robots, including: under-actuation of the floating-base joints, the rope pulling effect and the passive connection between the robot body and the rope *master-point*. These characteristics render the study of a rope-assisted bipedal robot both fascinating and unexplored, raising motion planning challenges when attempting to plan dynamic suspended maneuvers, as rappelling. To this end, we first introduce a template three-mass model of a bipedal robot connected through passive joints to an extensible rope, which is in turn modeled as a two-mass body. Based on this, a family of optimal control problems is presented to plan rappelling maneuvers.

I. INTRODUCTION

The Darpa Robotics Challenge (DRC) primarily aimed at using humanoid robotics technology to enter dangerous human-conceived environments, successfully encouraging the development of complex legged systems to be employed in disaster scenarios. The humanoid research community has consequently started to explore novel techniques to enhance and extend the capabilities of humanoids and legged systems. In [1] a control framework is presented to push a heavy object (more than 120 kg) with a quadrupedal humanoid robot. Boston Dynamics showed the Atlas robot performing the world-famous back flip and parkour demonstrators, although theoretical nor practical details have not been completely disclosed. Note also that DRC assessed the danger at environment-level rather than at task-level. The staged tasks in fact, although performed in a dangerous environment, were comparatively simple and safe for a human operator in nominal working conditions. As a step forward, in the near future we can imagine humanoid robots facing tasks with an inherent level of danger for the human worker. *Rope-assisted operations* such as: rescue, repair, maintenance and autonomous inspection *in the heights*, fall in this category [2]. At the present time, these tasks are entirely performed by human workforce, involving highly trained operators with expert decision making skills working in high-risk and complex environments, see Fig. 1. In this respect, the future employment of rope-assisted climbing humanoids could help in reducing human intervention in the heights, thus improving operators' health and safety.

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant No. 779963 (EUROBENCH).

All the authors are with the Humanoids and Human Centered Mechanics (HHCM) Lab., Istituto Italiano di Tecnologia, Genoa, Italy e-mail: {enrico.mingo, matteo.parigi, arturo.laurenzi, nikos.tsagarakis}@iit.it.



Fig. 1. 1: flying gardeners take care of the Bosco Verticale towers in Milan. 2: Rope-assisted fire-fighters saved the life of a survivor stuck in his vehicle several meters above the ground (Morandi bridge collapse, Genoa, August 14th, 2018). 3: Search & rescue operations in the heights and depths are carried out using rope-assisted techniques. 4: Wind turbine maintenance and inspection are field of application for robot climbing technologies.

Climbing robots have been addressed in the robotics literature during the last three decades and a wide variety of platforms have been proposed especially for vertical infrastructure inspection: Stickybot [3], SCAMP [4] or TAILS [5]. Unfortunately, the size and performance of these research platforms restrict their applicability merely to inspection tasks. Nevertheless, in recent years the readiness level of climbing robots technology has produced an increasing number of platforms for industrial applications [6]–[8]. However, dealing with *multiple kinematics chains*, e.g. a humanoid topology, in the control problem of a rope-assisted robotic system remains a comparatively unexplored research direction. Very few works can be found in the literature dealing with this topic. In [9] an impressive rappelling experiment has been carried on the HRP-2 bipedal humanoid. A simple strategy has been employed, consisting in a transition motion for rappelling, with a sequential transition of the Center of Mass (CoM) position and the contact states of the robot's limbs, while the rope is manipulated by both hands. From the same research group, a strategy to let HRP-2 walk on a steep slope has been presented in [10]. Here a Quadratic Programming (QP) problem is used to find a reference tension for the rope, while the locomotion part is left to a classical preview control approach [11]. In this regards, note that, despite both [9], [10] present remarkable experimental results, planning of rope-assisted dynamic motions have not been addressed so far.

Based on these considerations, the main contribution of this work are a comprehensive description of the floating-based dynamics of a rope-assisted bipedal system, based

on an introduced template model, and a family of Optimal Control problems (OCP), relying on the introduced template model, to plan rope-assisted rappelling maneuvers.

II. MATHEMATICAL MODEL

This Section is dedicated to the derivation of the equation of motion that describes the dynamics of a template model, which simplifies the kinematics of a rope-assisted bipedal robot, modeled as a three-mass body with point feet, connected through passive joints to an extensible rope, in turn modeled as a two-mass body. A similar *template* model can be found in [9] and relies on the assumption that rope-assisted manoeuvres, engage the motion of the robot legs solely, while the robot arms are used for in-place manipulation. Note that no further simplification will be introduced on the robot dynamic model.

A. Template Model for a Rope-Assisted Bipedal Robot

With reference to Fig. 2, let us consider a floating-base system consisting of the actuated *prismatic* joint coordinates $q_{ll} \in \mathbb{R}^3$ and $q_{lr} \in \mathbb{R}^3$, enabling the positioning of the left and right leg end-effectors, respectively. Assuming point contacts, the corresponding Cartesian positions and contact forces, expressed w.r.t. the world frame \mathcal{W} , are denoted with $p_{Cl}(q) \in \mathbb{R}^3$, $F_{Cl} \in \mathbb{R}^3$ and $p_{Cr}(q) \in \mathbb{R}^3$, $F_{Cr} \in \mathbb{R}^3$, respectively. Three additional *revolute* joints $q_b \in \mathbb{R}^3$ are introduced to model the passively-driven spherical connection between the robot body and the rope *master-point*, whose position w.r.t the world frame \mathcal{W} is denoted with $p_b(q) \in \mathbb{R}^3$. The rope itself is modeled as a two-mass system, where two thin rigid bodies are connected by a prismatic joint $q_r \in \mathbb{R}$, enabling the rope extension/contraction. The position of the rope *anchor-point* w.r.t. the world frame \mathcal{W} is denoted with $p_r(q) \in \mathbb{R}^3$ and the related contact force with $F_r \in \mathbb{R}^3$. The pose of the (under-actuated) floating-base is modeled through three prismatic joints $p_u \in \mathbb{R}^3$ and a spherical joint, whose orientation is given by the unit quaternion $\rho_u \in \mathbb{R}^4$. As shown in Fig. 2, the inertial properties of each rigid body are given by the related mass value and inertia tensor, while $p_{CoM} \in \mathbb{R}^3$ represents the position of the CoM w.r.t. the world frame \mathcal{W} . Finally, let us consider a *climbing surface* \mathcal{S} , whose orientation is described through its normal $n_S \in \mathbb{R}^3$. By denoting with n and n_u the number of actuated and unactuated degrees-of-freedom (DoFs), respectively, the generalized coordinates can be now collected in the vector $q \in \mathbb{R}^{n+n_u}$, with $n = 7$ and $n_u = 10$ for the introduced template model:

$$q = [p_u^T \quad \rho_u^T \quad q_{ll}^T \quad q_{lr}^T \quad q_b^T \quad q_r]^T, \quad (1)$$

while the generalized coordinates velocities $\nu \in \mathbb{R}^{n+n_u-1}$ and accelerations $\dot{\nu} \in \mathbb{R}^{n+n_u-1}$ are given by:

$$\nu = [\dot{p}_u^T \quad \omega_u^T \quad \dot{q}_{ll}^T \quad \dot{q}_{lr}^T \quad \dot{q}_b^T \quad \dot{q}_r]^T, \quad (2a)$$

$$\dot{\nu} = [\ddot{p}_u^T \quad \dot{\omega}_u^T \quad \ddot{q}_{ll}^T \quad \ddot{q}_{lr}^T \quad \ddot{q}_b^T \quad \ddot{q}_r]^T, \quad (2b)$$

where $\dot{p}_u, \ddot{p}_u \in \mathbb{R}^3$ and $\omega_u, \dot{\omega}_u \in \mathbb{R}^3$ are the linear and angular velocity and acceleration, respectively, of the robot floating-base expressed in the world \mathcal{W} coordinates.

Being $\rho_u = [\epsilon_u^T \quad \eta_u]^T$, with $\epsilon_u \in \mathbb{R}^3$ and $\eta_u \in \mathbb{R}$, the so-called *quaternion propagation* [12] is given by

$$\dot{\rho}_u = \left[\frac{1}{2} \omega_u, \quad 0 \right] \circ \rho_u, \quad (3)$$

expressing the relation between $\dot{\rho}_u$ and ω_u . The symbol \circ is used to denote the quaternion product.

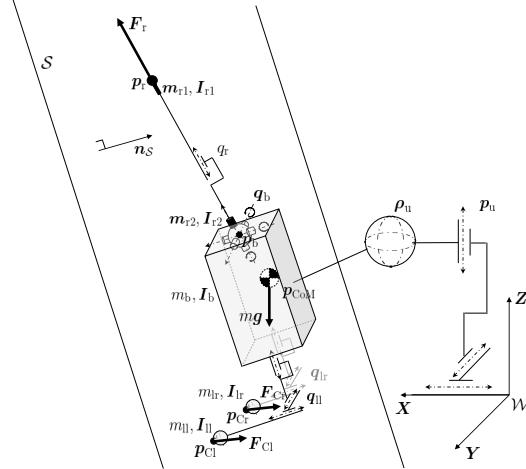


Fig. 2. The rope-assisted bipedal robot floating-base template model. The point p_r is the *anchor-point* between the rope and the surface \mathcal{S} . The connection between the rope and the body of the robot, modeled through the three rotational joints q_b is called *master-point* p_b .

B. Rope-Assisted Floating-Base Robot Dynamics

The dynamics of a rope-assisted floating-base robot can be expressed by the following equation of motion:

$$B(q)\dot{\nu} + h(q, \nu) = S\tau + J_C^T(q)F_C + J_r^T(q)F_r, \quad (4)$$

where $\tau = [\tau_{ll}^T, \tau_{lr}^T, \tau_r]^T \in \mathbb{R}^n$ are the actuated joint torques, while $B(q) \in \mathbb{R}^{(n+n_u-1) \times (n+n_u-1)}$ is the full joint space inertia matrix and $h(q, \nu) \in \mathbb{R}^{n+n_u-1}$ is the vector of non-linear (gravity, centrifugal/Coriolis) terms. Differently from a fixed-base robot, the following *actuation matrix* $S \in \mathbb{R}^{(n+n_u-1) \times n}$ is introduced to model the system under-actuation, consisting of the robot floating-base with the addition of the passively-driven master-point joints:

$$S = [0_{6 \times n} \quad I_{6 \times n} \quad 0_{3 \times n} \quad I_{1 \times n}]^T. \quad (5)$$

Contact forces $F_C \in \mathbb{R}^k$ are taken into account by concatenating the Jacobian of all support links $J_C(q) \in \mathbb{R}^{k \times (n+n_u-1)}$ and the corresponding overall contact wrench. We will hereafter assume n_C point contacts¹, thus $k = 3n_C$. The equation of motion in (4) can be further split into n_u unactuated and n actuated rows, denoted with subscript u and a , respectively:

$$B_u(q)\dot{\nu} + h_u(q, \nu) = J_{C,u}^T(q)F_C + J_{r,u}^T(q)F_r, \quad (6a)$$

$$B_a(q)\dot{\nu} + h_a(q, \nu) = \tau + J_{C,a}^T(q)F_C + J_{r,a}^T(q)F_r, \quad (6b)$$

¹For the case of surface contacts, wrenches exerted on each contacting link can be fully described by applying contact forces at the vertices of the contact polygon, which is consistent with the formulation given in (4).

with $\mathbf{J}_{C,u}^T(\mathbf{q}) \in \mathbb{R}^{(n_u-1) \times k}$ and $\mathbf{J}_{r,u}^T(\mathbf{q}) \in \mathbb{R}^{(n_u-1) \times 3}$.

In its current formulation, the floating-base model in (4), which is actually equivalent to the one of a floating-base tripod system, does not fully capture the rope-assisted dynamics. Therefore, in order to properly model the rope suspension effects, the following set of additional constraints needs to be applied to the equation of motion in (4).

1) Rope constraints: We have already modeled the rope *master-point* by introducing a ball joint \mathbf{q}_b and by making it passive thanks to the actuation matrix \mathbf{S} in (5). To properly model the effects of the rope suspension, we now need to first include a constraint on the position of the *anchor-point* $\mathbf{p}_r(\mathbf{q})$ to be equal to a prescribed position $\mathbf{p}_r^{\text{des}}$, i.e.:

$$\mathbf{p}_r(\mathbf{q}) = \mathbf{p}_r^{\text{des}}. \quad (7)$$

It is common practice to employ a linearized version of (7) w.r.t. joint velocities and/or accelerations:

$$\dot{\mathbf{p}}_r(\mathbf{q}, \boldsymbol{\nu}) = \mathbf{J}_r(\mathbf{q})\boldsymbol{\nu} = \mathbf{0}, \quad (8a)$$

$$\ddot{\mathbf{p}}_r(\mathbf{q}, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}}) = \mathbf{J}_r(\mathbf{q})\dot{\boldsymbol{\nu}} + \dot{\mathbf{J}}_r(\mathbf{q}, \boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{0}. \quad (8b)$$

Finally, the pulling force \mathbf{F}_r applied by the rope on the *anchor-point* $\mathbf{p}_r(\mathbf{q})$ can be implicitly constrained by the torque $\tau_r \in \mathbb{R}$ exerted on the rope prismatic joint q_r . In particular, by imposing the following unilateral constraint:

$$\tau_r \leq 0, \quad (9)$$

we can ensure that the rope can only pull from the anchor-point and not push.

2) Climbing surface constraints: When the robot is establishing contact with the climbing surface \mathcal{S} , constraints on contact positions and contact forces must be simultaneously enforced. The former type of constraint, i.e.:

$$\mathbf{p}_{Ci}(\mathbf{q}) \in \mathcal{S}(\mathbf{p}_{Ci}(\mathbf{q})), \quad (10)$$

ensures that the i -th contact point, with $i := \{l, r\}$, lays on the surface. In the simple planar case, the climbing surface equation is given by:

$$\mathcal{S}(\mathbf{p}_{Ci}(\mathbf{q})) : \mathbf{n}_S^T \mathbf{p}_{Ci} + d = 0, \quad (11)$$

being $\mathbf{n}_S = [a \ b \ c]^T \in \mathbb{R}^3$ the climbing surface normal, with $a, b, c, d \in \mathbb{R}$. We also need to ensure that contact points do not slip on the climbing surface, i.e.:

$$\dot{\mathbf{p}}_{Ci}(\mathbf{q}, \boldsymbol{\nu}) = \mathbf{J}_{Ci}\boldsymbol{\nu} = \mathbf{0}. \quad (12)$$

In order to further encode the impact of the climbing surface orientation on contact forces, friction constraints must be incorporated. Let us consider the contact force $\mathbf{F}_{Ci} \in \mathbb{R}^3$ at the i -th contact point, with $i := \{l, r\}$. Being $\mathbf{F}_{Ci}^n \in \mathbb{R}^3$ and $\mathbf{F}_{Ci}^t \in \mathbb{R}^3$, the normal and tangential component of the contact force, respectively, given by:

$$\mathbf{F}_{Ci}^n = (\mathbf{F}_{Ci} \cdot \mathbf{n}_S)\mathbf{n}_S, \quad (13a)$$

$$\mathbf{F}_{Ci}^t = \mathbf{F}_{Ci} - (\mathbf{F}_{Ci} \cdot \mathbf{n}_S)\mathbf{n}_S, \quad (13b)$$

the i -th point contact remains in rest contact mode if \mathbf{F}_{Ci} lies inside the friction cone directed by \mathbf{n}_S , i.e.:

$$\mathcal{F}(\mathbf{F}_{Ci}, \mathbf{n}_S, \mu) := \begin{cases} \mathbf{F}_{Ci} \cdot \mathbf{n}_S > F_{\text{thr}} \\ \|\mathbf{F}_{Ci}^t\|_2 \leq \mu(\mathbf{F}_{Ci} \cdot \mathbf{n}_S), \end{cases} \quad (14)$$

where μ is the Coulomb friction coefficient, while $F_{\text{thr}} \geq 0$ is a scalar force threshold. The Euclidean norm $\|\cdot\|_2$ models Coulomb friction cones with circular section. A largely adopted linearized version of (14) is given by:

$$\underline{\mathbf{b}} \leq \mathbf{D} \mathbf{R}_S \mathbf{F}_C \leq \bar{\mathbf{b}}. \quad (15)$$

The matrix $\mathbf{R}_S \in \mathbb{R}^{3 \times 3}$ maps the contact forces in a local reference frame. One can choose \mathbf{R}_S such that the z -axis represents the normal component of the contact force. Being $\{\text{C}, i\} \mathbf{F}_{Ci} = \mathbf{R}_S \mathbf{F}_{Ci} \in \mathbb{R}^3$ the i -th contact force expressed w.r.t. the local reference frame $\{\text{C}, i\}$, the matrix \mathbf{D} and the lower and upper bounds $\underline{\mathbf{b}}, \bar{\mathbf{b}}$, respectively, can be retrieved from the expression of the polyhedral linearized friction cones.

III. OCP FORMULATION AND TRANSCRIPTION

In this Section we introduce the general OCP which employs the dynamic model in Sec. II to plan rope-assisted rappelling motions. The OCP will be first formulated in continuous time and subsequently transcribed using a *Direct Multiple Shooting* (DMS) discretization [13].

Let us first consider the following choice for the state \mathbf{x} and control \mathbf{u} vectors:

$$\mathbf{x} = [\mathbf{q}^T \ \boldsymbol{\nu}^T]^T, \quad (16a)$$

$$\mathbf{u} = [\dot{\boldsymbol{\nu}}^T \ \mathbf{F}_C^T \ \mathbf{F}_r^T]^T, \quad (16b)$$

hence: $\dot{\mathbf{x}} = [\dot{\mathbf{q}}^T \ \dot{\boldsymbol{\nu}}^T]^T$. Based on the floating-base model described in Sec. II, the OCP formulation we address in this Section reads as:

$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_0^T L(\mathbf{x}(t), \mathbf{u}(t)) dt + E(\mathbf{x}(T))$ subject to $\mathbf{x}_0 - \mathbf{x}^{\text{init}} = \mathbf{0}$ $\mathbf{x}(T) - \mathbf{x}^{\text{goal}} = \mathbf{0}$	initial state final state
$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0}$	double integrator
$\tau_u(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0}$	under-actuation (6a)
$\underline{\tau} \leq \tau(\mathbf{x}(t), \mathbf{u}(t)) \leq \bar{\tau}$	torque bounds (6b)
$\mathbf{p}_r(t) - \mathbf{p}_r^{\text{des}} = \mathbf{0}$	rope anchor-point (7)
$\tau_r \leq 0$	rope pull (9)
$\underline{\mathbf{p}}_{Ci} \leq \mathbf{p}_{Ci}(t) \leq \bar{\mathbf{p}}_{Ci}$ if Ci in contact:	work-space bounds
$\mathbf{p}_{Ci}(t \in T_C) \in \mathcal{S}(\mathbf{p}_{Ci})$	climbing surface (10)
$\dot{\mathbf{p}}_{Ci}(t \in T_C) = \mathbf{0}$	no slip condition (12)
$\mathbf{F}_{Ci}(t \in T_C) \in \mathcal{F}(\mathbf{F}_{Ci}, \mathbf{n}_S, \mu)$	friction cone (14)
otherwise:	
$\mathbf{F}_{Ci}(t \notin T_C) = \mathbf{0}$	no contact force

Herein the *double integrator* relation from $\dot{\nu}(t)$ to $q(t)$ can be expressed through the following state-space representation:

$$\dot{x}(t) = \begin{bmatrix} \mathbf{0} & \hat{S} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} x(t) + f_{\text{quat}}(x(t)) + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} & \mathbf{0} \end{bmatrix} u(t), \quad (18)$$

where the selection matrix \hat{S} is given by:

$$\hat{S} = [I_{3 \times (n+n_u)} \quad \mathbf{0}_{4 \times (n+n_u)} \quad I_{(n+3) \times (n+n_u)}]^T \quad (19)$$

while $f_{\text{quat}}(x)$ represents the quaternion propagation, previously introduced in (3). The under-actuation constraint is implemented according to (6a) by denoting with $\tau_u \in \mathbb{R}^{n_u-1}$ the under-actuated torques. Similarly, torque bounds are imposed on the actuated torques $\tau \in \mathbb{R}^n$, whose expression can be retrieved from (6b). The rope pulling effect has been enforced as a constraint in (9). Bounds on the contact positions are introduced to approximate the reachable workspace through box constraints.

In order to perform a DMS transcription [13] of (17) into a Nonlinear Program (NLP) that can be solved by off-the-shelf solvers, let us consider a number N_s of shooting intervals, which discretize the control horizon. The state variable and control vector at the k -th shooting interval, x_k and u_k respectively, are denoted as:

$$x_k = [q_k^T \quad \nu_k^T]^T \quad (20a)$$

$$u_k = [\dot{\nu}_k^T \quad F_{C,k}^T \quad F_{r,k}^T]^T \quad (20b)$$

We hereafter assume a piece-wise constant control parametrization along each shooting interval. The $N_s + 1$ states and N_s controls are collected in:

$$X = [x_0^T, x_1^T, \dots, x_{N_s}^T]^T \quad (21a)$$

$$U = [u_0^T, u_1^T, \dots, u_{N_s-1}^T]^T \quad (21b)$$

In agreement with DMS, the so-called “continuity condition” constraint needs to be further satisfied: $s(x_k, u_k) - x_{k+1} = 0$. Here the function $s(x_k, u_k)$ is used to simulate the double-integrator dynamics in (18) over one shooting interval. We additionally let the solver free to decide the optimal step size for each shooting interval, introducing the step size variable $dt_k \in \mathbb{R}^+$ in the control vector:

$$u_k = [\dot{\nu}_k^T \quad F_{C,k}^T \quad F_{r,k}^T \quad dt_k]^T, \quad (22)$$

together with the positive bound: $\underline{dt} \leq dt_k \leq \bar{dt}$. Finally, we adopted a linearized version of friction cones which, in our experience, is easier to handle for the solver.

IV. PLANNING OF ROPE-ASSISTED MANEUVERS

This Section is dedicated to the planning problem of rope-assisted maneuvers requiring direct interaction with the environment, hereafter represented by a vertical wall. In particular we consider a controlled descent off the climbing wall, technically referred to as *rappelling*. The related NLPs have been implemented using the CasaADi [14] library, an

open-source tool for nonlinear optimization and automatic differentiation. The robot model described in Sec. II has been used within CasADi thanks to Pinocchio library [15], providing analytical derivatives of the main rigid-body algorithms, e.g. the recursive Newton-Euler algorithm or the articulated-body algorithm. Mass and inertial parameters of the template model have been chosen according to the ones of the real size bipedal humanoid COMAN+ [16]: 40 kg for the base, 10 kg for the feet, with inertia approximated as a box and spheres respectively. Rope mass has been chosen as 1 kg.

Rappelling

The rappelling is a technique used by climbers, and rope access technicians to descend cliffs or slopes when they are too steep and/or dangerous to descend. Inspired by the work of Winkler et al. [17] for legged locomotion, in the considered scenario the rappelling has been characterized by four *actions* denoted with (L, R, D, F) , where L : left foot in contact, R : right foot in contact, D : both feet standing, F : flight phase. Note that both jumps and stepping motions are inherently considered in this characterization, while the rope extension can be actively modified. A user-defined number of rappelling-*actions* (n_a) is scheduled within a single rappelling-*phase*. Multiple *phases* (n_p) can be considered, while each *action* is applied for N_a shooting intervals in order to let the solver modify its duration (see Fig. 5). Subsequently, the total number of shooting intervals N_s needs to fulfill the relation $N_s \geq n_p \cdot n_a \cdot N_a$. This choice has been driven by the employment of a DMS discretization, whereas in [17] *Direct Collocation* (DC) [13] is adopted. DC does not require an integrator of the system dynamics, trading-off computational advantage against dynamical accuracy. Differently from DMS, this enables the possibility to optimize each action duration *concurrently* [17]. For this reason, we here propose to optimize each action duration *sequentially*, see Fig. 5, in order to extend the approach in [17] to DMS discretization. We can now set up different types of rappelling descents along the vertical climbing surface. According to [9], in these scenarios the *master-point* location has been shifted towards the center of the robot body in order to prevent it from tilting excessively towards the climbing wall.

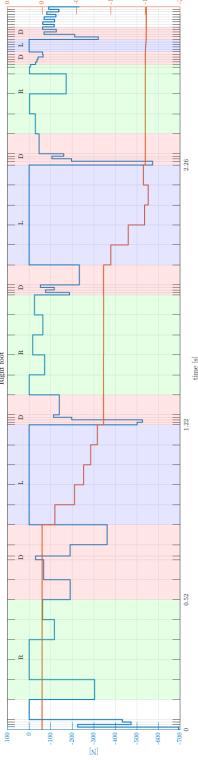
1) *Jumping strategy*: By choosing as candidate actions a double stance D followed by a flight phase F (see Table II), it is possible to rappel down the wall using a *jumping* strategy. The following cost function has been considered:

$$\begin{aligned} F(X, U) = & \sum_{k=0}^{N_s} \left(\gamma_\nu \nu_k^T \nu_k + \gamma_{p_u} (p_{u,k}^{[x]} - p_{u,k}^{[x],\text{ref}})^2 \right) + \\ & + \sum_{k=0}^{N_s} \left(\gamma_{\rho_u} (\rho_{u,k} - \rho_u^{\text{ref}})^T (\rho_{u,k} - \rho_u^{\text{ref}}) + \gamma_{q_r} (q_{r,k} - q_r^{\text{goal}})^2 \right), \end{aligned} \quad (23)$$

with $\gamma_\nu = 1$, $\gamma_{p_u} = 100$, $\gamma_{\rho_u} = 100$, $\gamma_{q_r} = 10$. Here $q_r^{\text{goal}} = 3$ m is the desired length of the rope, while $p_u^{[x],\text{ref}}$ is used to jump at a certain distance from the wall and ρ_u^{ref} to keep upright the floating-base orientation. Fig. 3.a

TABLE III
NLP SPECIES FOR RAPPELLING WITH STEPPING STRATEGY (L: LEFT FOOT IN CONTACT, R: RIGHT FOOT IN CONTACT, D: BOTH FEET STANDING, F: FLIGHT PHASE, n_a : NO. OF ACTIONS, N_a : NO. OF PHASES, n_p : NO. OF PHASERS, N_p : SHOOTING INTERVALS PER ACTION, N_s : SHOOTING INTERVALS)

Stepping	(R, D, L, D)	4	5	3	75	0.01 s	0.08 s	2.90 s	4399	5044	1200
Strategy	Actions	n_a	N_a	n_p	N_p	n_s	\bar{dt}	Horizon	Variables	Constraints	Max iterations



a. Time history of the contact forces z -component (blue solid line) and Cartesian position of the foot along the z -axis (red solid line) during the rappelling using a stepping strategy, see Table III. The separate actions are represented by distinct colors. It can be noted how the solver automatically decides not to perform the last jump, reducing the duration of the last D and F actions.

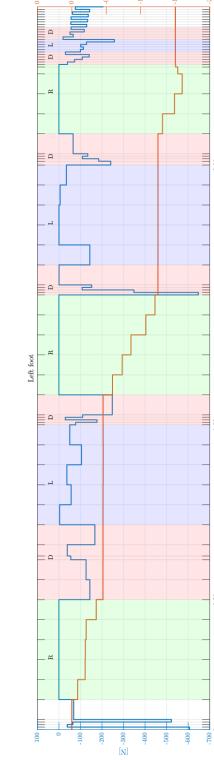
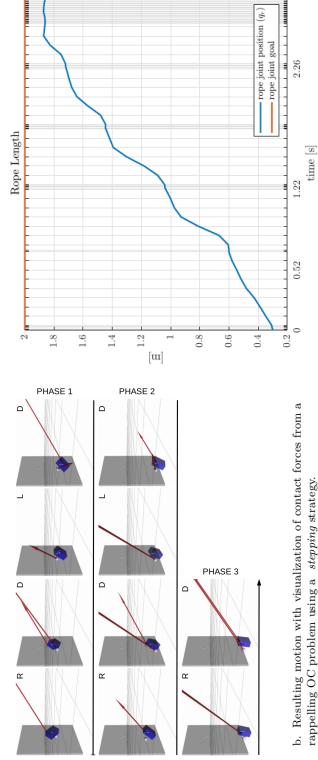
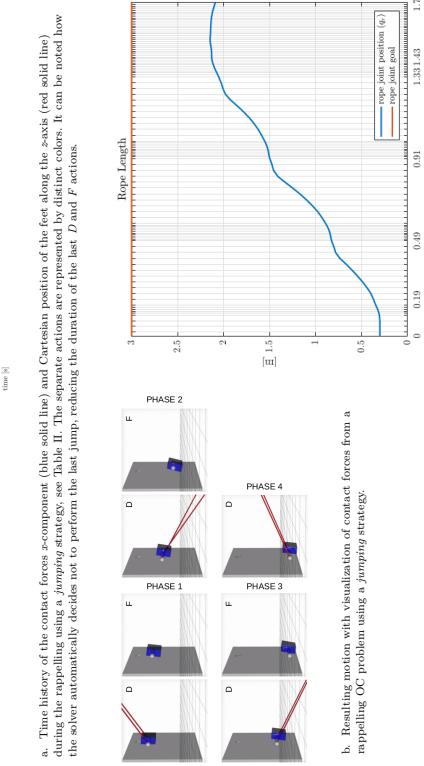
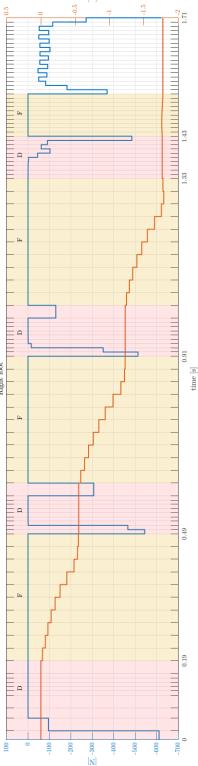


Fig. 4. Rappelling with *stepping* strategy.



b. Resulting motion with visualization of contact forces from a rappelling OC problem using a *stepping* strategy.
c. Time history of the rope joint position (blue solid line) and rope target position (red solid line) from a rappelling using a *stepping* strategy.



a. Time history of the contact forces z -component (blue solid line) and Cartesian position of the foot along the z -axis (red solid line) during the rappelling using a jumping strategy, see Table II. The separate actions are represented by distinct colors. It can be noted how the solver automatically decides not to perform the last jump, reducing the duration of the last D and F actions.

b. Resulting motion with visualization of contact forces from a rappelling OC problem using a *jumping* strategy.
c. Time history of the rope joint position (blue solid line) and rope target position (red solid line) from a rappelling using a *jumping* strategy.

TABLE II
NLP SPECIES FOR RAPPELLING WITH JUMPING STRATEGY (L : LEFT FOOT IN CONTACT, R : RIGHT FOOT IN CONTACT, D : BOTH FEET STANDING, F : FLIGHT PHASE, n_a : NO. OF ACTIONS, N_a : NO. OF PHASES, n_p : SHOOTING INTERVALS PER ACTION, N_s : SHOOTING INTERVALS)

Jumping	(D, F)	2	10	4	100	0.01 s	0.03 s	1.74 s	5874	6486	4000
Strategy	Actions	n_a	N_a	n_p	N_p	n_s	\bar{dt}	Horizon	Variables	Constraints	Max iterations

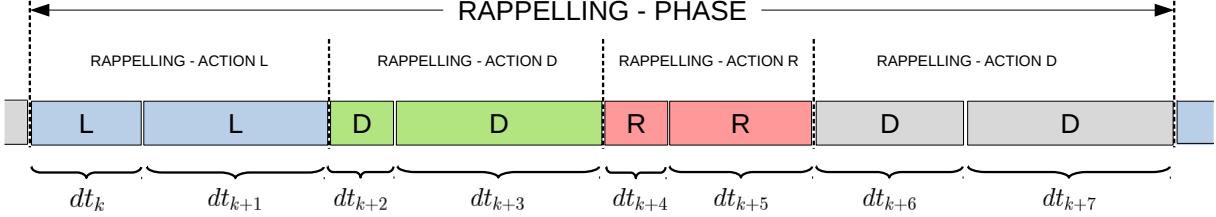


Fig. 5. A *phase* is characterized by multiple consecutive *rappelling-actions*. The time of duration of a *rappelling-action* is as well a variable of the NLP. During a *rappelling-action* the control action of the solver is piece-wise constant. To allow the solver to change control action during the same *rappelling-action*, is possible to split the latter over multiple shooting nodes.

shows the time histories of the normal component of the contact forces (x -axis) on each foot and the related foot position along the z -axis. As it can be noticed, during the flight phases, the feet move along the z -axis due to the rope extension towards the goal q_r^{goal} (see Fig. 3.c), while no contact force is applied. Note also that, although the number of user-defined *phases*, i.e. the number of jumps, is equal to 4 (see Table II), the solver automatically decides not to perform the last jump. A double stance D action is applied in the remaining shooting intervals (white areas in Fig. 3.a). Snapshots from the produced motion are depicted in Fig. 3.b.

2) *Stepping strategy*: The same vertical descent task can be performed considering a sequence of single and double stance actions, as summarized in Table III, according to a *stepping* strategy. The same cost function in (23) has been adopted with $\gamma_v = 1$, $\gamma_{p_u} = 1$, $\gamma_{\rho_u} = 1$, $\gamma_{q_r} = 1$ and $q_r^{\text{goal}} = 2 \text{ m}$. The time histories of the normal component of the contact forces x -component, i.e. normal to the climbing wall, and the related foot position along the z -axis are shown in Fig. 4.a. Note that the duration of the single stance actions (L) during the last phase is reduced to the minimum, showing how the solver is free to choose the number of steps. The sequence in Table III is preceded and followed by a double stance D action (white areas in Fig. 4.a). The time history of the rope joint evolution q_r is shown in Fig. 4.c, while snapshots from the produced motion are depicted in Fig. 4.b. Note that by properly modifying q_r^{goal} in (23) together with the initial state x^{init} , an ascending *climbing* motion can be alternatively produced. For brevity reasons this result is omitted here and shown in the accompanying video.

3) *Gap crossing*: A gap crossing example can be further considered by describing the vertical climbing wall as follows:

$$S(p_{Ci}) := \begin{cases} p_{Ci}^{[x]} - x_{\text{wall}} = 0 \\ \epsilon_{\text{gap}}(p_{Ci}^{[z]} - z_{\text{gap}})(p_{Ci}^{[z]} - \bar{z}_{\text{gap}}) \geq 0. \end{cases} \quad (24)$$

Here, a 1 m wide gap has been modeled as a parabolic function with $\bar{z}_{\text{gap}} = -1.5 \text{ m}$ and $z_{\text{gap}} = -2.5 \text{ m}$, where $\epsilon_{\text{gap}} \in \mathbb{R}$ is a relaxation factor, which in our experience helps the NLP solver to converge to a solution. For this example a sequence of both *stepping* and *jumping* actions has been employed, in order to let the solver free to decide whether to cross the gap with a step or a jump. To this

end, the same cost function in (23) has been considered, where the last term has been replaced with the following: $\gamma_{p_C}(p_{C,k}^{[z]} - p_C^{[z],\text{goal}})^T(p_{C,k}^{[z]} - p_C^{[z],\text{goal}})$ in order to track at best a 4 m downward displacement for each foot. Here $\gamma_v = 0.1$, $\gamma_{p_u} = 2$, $\gamma_{\rho_u} = 100$, $\gamma_{p_C} = 10^3$. The algorithm is able to effectively transverse the gap, automatically choosing to perform a jump. This is consistent with the introduced work-space constraints on the feet positions, which prevent the robot from stepping across the gap. The algorithm is therefore able to automatically discover the gait pattern, changing the initially provided gait sequence and timings depending on the climbing surface and desired task. Again, for brevity reasons, simulations results are reported in the accompanying video.

V. CONCLUSIONS

This letter has addressed the mathematical modeling of a template three-mass model connected through passive joints to an extensible rope. The presented dynamic model has been employed within a family of OCPs in order to plan rope-assisted maneuvers. The effectiveness of the proposed approach has been especially validated in a series of rappelling scenarios, where an automatic discovery of the gait pattern has been enabled. The demonstration of the planned motions in a real bipedal platform requires a controller that is able to reliably track the generated trajectories, incorporating sensor data in order to calculate the appropriate joint positions and/or torques. Building upon authors' previous work [1], [18], future work will introduce a control framework capable to transform planned contact forces and Cartesian references, into proper joint level commands for the real robot. To this end, the effect of the rope will be incorporated in the control framework, together with estimation of the status of the passive *master-point* joints. Experimental validation will also require technological effort to realize a dedicated infrastructure to perform experiments. On a final note, it is the authors belief that the topics considered in this work will gain the attention of the humanoid community in the forthcoming years.

REFERENCES

- [1] M. Parigi Polverini, A. Laurenzi, E. Mingo Hoffman, F. Ruscelli, and N. G. Tsagarakis, "Multi-Contact Heavy Object Pushing with a Centaur-Type Humanoid Robot: Planning and Control for a Real Demonstrator," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859–866, 2020.

- [2] V. Sudevan, A. Shukla, and H. Karki, "Current and future research focus on inspection of vertical structures in oil and gas industry," in *IEEE International Conference on Control, Automation and Systems*, 2018, pp. 144–149.
- [3] S. Kim, M. Spenko, S. Trujillo, B. Heyneman, D. Santos, and M. R. Cutkosky, "Smooth vertical surface climbing with directional adhesion," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 65–74, 2008.
- [4] M. T. Pope, C. W. Kimes, H. Jiang, E. W. Hawkes, M. A. Estrada, C. F. Kerst, W. R. T. Roderick, A. K. Han, D. L. Christensen, and M. R. Cutkosky, "A multimodal robot for perching and climbing on vertical outdoor surfaces," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 38–48, 2017.
- [5] M. P. Austin, M. Y. Harper, J. M. Brown, E. G. Collins, and J. E. Clark, "Navigation for legged mobility: Dynamic climbing," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 537–544, 2019.
- [6] K. Berns, C. Hillenbrand, and T. Luksch, "Climbing robots for commercial applications—a survey," in *International Conference on Climbing and Walking Robots*, 2003, pp. 17–19.
- [7] "Wall Robotic Technologies," <https://www.wallrobotic.com/>.
- [8] "Rope Robotics," <http://roperobotics.com/>.
- [9] M. Bando, M. Murooka, T. Yanokura, S. Nozawa, K. Okada, and M. Inaba, "Rappelling by a humanoid robot based on transition motion generation and reliable rope manipulation," in *IEEE-RAS International Conference on Humanoid Robotics*. IEEE, 2017, pp. 129–135.
- [10] M. Bando, M. Murooka, S. Nozawa, K. Okada, and M. Inaba, "Walking on a steep slope using a rope by a life-size humanoid robot," in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 705–712.
- [11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, pp. 1620–1626.
- [12] B. Graf, "Quaternions and dynamics," 2008.
- [13] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.
- [14] 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, no. 1, pp. 1–36, 2019.
- [15] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations*, 2019.
- [16] E. Mingo Hoffman, M. Parigi Polverini, A. Laurenzi, and N. G. Tsagarakis, "A study on sparse hierarchical inverse kinematics algorithms for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 235–242, Jan 2020.
- [17] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.
- [18] F. Ruscelli, M. Parigi Polverini, A. Laurenzi, E. Mingo Hoffman, and N. G. Tsagarakis, "A multi-contact motion planning and control strategy for physical interaction tasks using a humanoid robot," in *IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2020.