Locomotion Adaptation in Heavy Payload Transportation Tasks with the Quadruped Robot CENTAURO

Xinyuan Zhao[1,2], Yangwei You[3], Arturo Laurenzi[1], Navvab Kashiri[1] and Nikos Tsagarakis[1]

***Abstract*— This paper presents a reactive legged locomotion generation scheme that enables our quadruped robot CENTAURO to adapt to varying payloads while walking. The center-of-mass (CoM) trajectories are generated in real time in a model predictive control (MPC) fashion, trading off large stability margins against evenly stretched legs. Vertex-based zero-moment-point (ZMP) constraints are imposed to ensure quasi-static walking stability. A Kalman filter is then implemented to estimate the CoM states and the impact of external payloads which can vary online and affect/disturb the locomotion differently. The CoM estimation is used to update the MPC motion planner at every replanning instant so that the robot can react to unknown and time-varying payloads on the fly.**

We validate the proposed scheme through experimental trials where the robot walks on flat ground or steps on different surface levels while carrying heavy payloads. It is shown that the proposed reactive locomotion strategy enables the robot to carry 20 kg payloads, which is close to the maximum capacity of the robot arms.

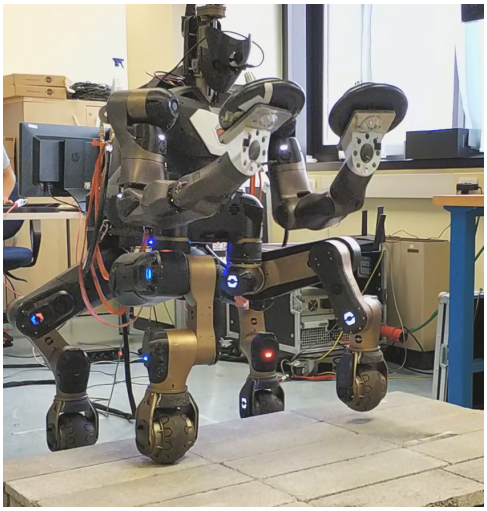

Fig. 1. CENTAURO, a wheeled-legged quadruped robot with two arms.

I. Introduction

CENTAURO is a wheeled-legged quadruped robot with a dual-arm upper body [1], [2], as shown in Fig. 1. Compared with other mobile manipulation robots, e.g. [3]–[5], it features significant physical capabilities that enable the robot to perform demanding tasks such as manipulation and transportation of heavy payloads, with a maximum of 12 kg payload on each arm and a 60 kg payload on its pelvis [2]. Different from legged/tracked robots, the hybrid mobility of CENTAURO, which combines articulated legs with wheels permits the robot to traverse terrains of difficult geometry.

One of the commonly adopted approaches in generating quasi-static walking locomotion is to have the zero-moment point (ZMP) always inside the support polygon (SP), which has been employed on both biped and quadruped robots [6]–[8]. When the robot is modeled as a linear inverted pendulum model (LIPM), the ZMP is typically calculated from the positions and accelerations of the center-of-mass (CoM). Herdt et al. [9] propose an MPC-based scheme for generation of bipedal walk. The CoM dynamics are captured by a LIPM, and the footholds are determined online together with the CoM motions by solving a single quadratic program (QP) problem. The walking stability is ensured by constraining the ZMP always lying inside the SP, which has been proved to be a necessary condition for LIPMs by Wieber [10]. Laurenzi et al. [11] develop an MPC-based framework to generate legged locomotion for quadruped robots with automatic footstep placement and explicit stability guarantee. A recent work of Sun et al. [12] presents a scheme that produces legged-wheeled hybrid locomotion for CENTAURO using MPC.

However, it is observed that the CoM position of CENTAURO could be shifted by about 10 cm from its nominal value when the robot carries 20 kg payloads by arms. In order to generate practically stable locomotion under this circumstance, it is necessary to estimate the real CoM states and update the motion planner online, see, e.g., [13]–[15]. When applying the locomotion generation scheme proposed in our previous works [11], [12], the robot is unaware of the effect caused by the payloads, and thus the walking stability could be compromised in payload-transportation tasks. To address payload-transportation tasks, [16] proposes to identify the inertial parameters of the trunk when a change of payloads is detected. Nevertheless, one restriction of this approach is that the identification can only be made when the robot is static. Orsolino et al. [17] propose a concept named *Feasible Region*, which represents a set of CoM positions that respect static equilibrium, friction constraints, and actuation feasibility. However, the combination of this approach with dynamic stability criteria such as ZMP constraints is not presented. The walking stability can only be implicitly ensured by taking slow motions. Therefore, we are motivated to develop a legged locomotion generation scheme, which can react and adapt to unknown payloads that may vary during the walk.

The main contribution of this work is therefore the

[1]Humanoid and Human Centered Mechatronics Research Line, Istituto Italiano di Tecnologia, via Morego 30, Genova, Italy, 16163. name.surname@iit.it

[2]DIBRIS, Universita di Genova, Italy, 16145.

[3]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore, 138632. youy@i2r.a-star.edu.sg

development and validation of an effective planning and control framework that generates adaptive legged locomotion to unknown and online-varying payloads for quadruped robots. It mainly consists of a Cartesian motion planner, a hierarchical whole-body controller and a CoM state estimator, which will be introduced later. Through comparative experimental studies in different scenarios, we demonstrate that it enables the robot to carry total of 20 kg online-varying payloads by the arms during walking.

The rest of this paper is organized as follows. An overview of the proposed scheme is given in Section II. In Section III, the CoM motion planner is derived, followed by the implementation of the CoM estimator and the whole-body controller. Simulations and experiments are presented in Section IV, where we compare and analyze the results of different locomotion generation strategies. Section V concludes the paper and presents possible future directions.

II. Overview

An overview diagram of the proposed scheme is displayed in Fig. 2.

Based on pre-defined gait patterns and user inputs, the CoM reference motions, including the position, velocity and acceleration trajectories, are generated by the MPC motion planner, where a vertex-based ZMP constraint is imposed to ensure quasi-static walking stability. To avoid pushing the legs to kinematic limits, a trade-off term is added to the cost function to balance the pursuit of large stability margins and evenly stretched legs. Trajectories of swing feet are formulated as fifth order polynomial functions, connecting the initial position, the top position indicated by pre-defined ground clearance, and the nominal touchdown position calculated by the reference velocity of the CoM. We adopt a fixed gait schedule that is commonly used to generate crawl locomotion in the trials: left-front leg (LF) $\rightarrow$ right-hind leg (RH) $\rightarrow$ right-front leg (RF) $\rightarrow$ left-hind leg (LH). Each stride is composed of a *swing phase* where one of the legs lifts and reaches out, and a *contact phase* where all the legs are in contact, with their periods denoted as t_s and t_c, respectively. Given the Cartesian reference motions, the whole-body controller solves a hierarchical QP optimization to compute the joint reference motions, which trades off among multiple objectives and complies with mechanical constraints of the actuators.

The disturbance of the unknown external payloads is evaluated by how far the real CoM deviates from the nominal values. A Kalman filter is thus implemented to estimate the CoM states and the deviations based on the inertial measurement unit (IMU) measurements and estimated center-of-pressure (CoP) positions. The CoM motion planner is then updated by the estimations at each replanning instant to generate reactive behaviors that compensate for the disturbance of the payloads.

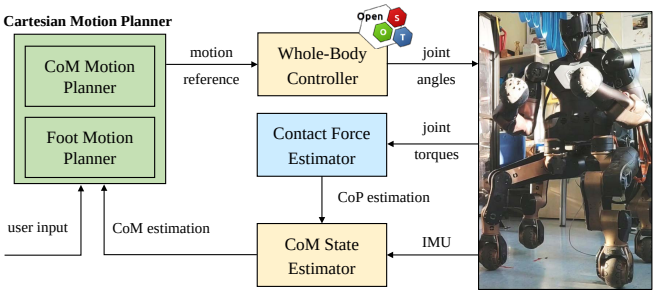

Fig. 2. An overview diagram of the proposed locomotion generation scheme. Blocks with green shades run at 10 Hz; blocks with yellow shades run at 100 Hz; and the block with blue shade runs at 200 Hz.

III. Motion Planning, Estimation and Control

A. CoM Motion Planner

The CoM trajectories are optimized in an MPC fashion where the robot is modeled as a LIPM. Let $\mathbf{c} = [c^x, c^y]$ and $\mathbf{z} = [z^x, z^y]$ denote the xy coordinates of the CoM and the ZMP in the world coordinate frame, respectively. The feasible motions of the CoM are connected to the ZMP by

$$\ddot{\mathbf{c}} = \omega^2(\mathbf{c} - \mathbf{z}) \tag{1}$$

where $\omega = \sqrt{g/h}$ and g is the gravitational acceleration constant, being 9.81 m/s^2, and h is the height of the CoM from the ground. A ZMP constriant is then imposed to guarantee quasi-static gaits

$$\mathbf{z} \in \text{ConvHull}\left\{\mathbf{p}_i\right\}, \quad i \in \mathbb{S} \tag{2}$$

where $\mathbf{p}_i = [p_i^x, p_i^y]$ is the xy coordinates of the i^{th} foot, and $\mathbb{S}$ is the set of indices of stance legs at that moment. Hence, ConvHull $\{\mathbf{p}_i\}$ represents the convex hull formed by the current footholds, i.e., the support polygon.

Following the same way as in [9], we discretize the continuous dynamics Eq. (1) by a constant time interval T, build up a discretized linear MPC with a predicting horizon of N steps, and transcribe it into a classical QP formulation

$$\min_{\mathbf{u}} \sum_{k=1}^{N} \frac{\alpha}{2}\|\dddot{\mathbf{c}}_k\|^2 + \frac{\beta}{2}\|\mathbf{z}_k - \mathbf{z}_k^{\text{ref}}\|^2 \tag{3}$$

where the decision variable $\mathbf{u} = [\dddot{\mathbf{c}}_1, \ldots, \dddot{\mathbf{c}}_N]$ is a collection of all the N CoM jerks, and $\mathbf{z}_k^{\text{ref}}$ denotes the desired position of the ZMP at time instant k.

The gain α in (3) plays the role to smoothen the optimized CoM motions. In this work, we choose the reference of the ZMP, i.e. $\mathbf{z}_k^{\text{ref}}$, to be the center of the support polygon at each time instant k. The β-related term attempts to minimize the deviation of the ZMP to its reference and thus contributes to maximizing the stability margins during walking. However, as what will be demonstrated in Section IV, merely pursuing large stability margins will push the robot to kinematic limits and thus hinder the locomotion. To alleviate this issue, we propose to add a third term to the original optimization (3) that seeks to modulate the legs equally, yielding

$$\min_{\mathbf{u}} \sum_{k=1}^{N} \frac{\alpha}{2}\|\dddot{\mathbf{c}}_k\|^2 + \frac{\beta}{2}\|\mathbf{z}_k - \mathbf{z}_k^{\text{ref}}\|^2 + \frac{\gamma}{2}\|\mathbf{c}_{p,k} - \mathbf{c}_{p,k}^{\text{ref}}\|^2 \tag{4}$$

where $\mathbf{c}_{p,k} = \left[c^x_{p,k}, c^y_{p,k}\right]$ represents the xy coordinates of the geometric center of the pelvis. $\mathbf{c}^{\text{ref}}_{p,k}$ is selected to be the geometric center of the convex hull formed by all feet at instant k regardless of their contact states. Therefore, the γ-related term aims to keep the pelvis close to the SP center where the legs are stretched equally. By tuning the gains β and γ in the optimization (4), we trade off larger stability margins against more evenly stretched legs. The vector from $\mathbf{c}_k$ to $\mathbf{c}_{p,k}$, denoted as $\mathbf{d}$, can be obtained from the kinematics model and is assumed to be constant within each optimization horizon. A quadratic optimization is then established by substituting $\mathbf{c}_{p,k} = \mathbf{c}_k + \mathbf{d}$ into (4).

With regard to the stability condition (2), we propose to adopt a vertex-based ZMP constraint, similar to [8], [18]. The constraint is fulfilled if and only if the ZMP is equal to a convex combination of all the supporting footholds, that is

$$\begin{aligned} &\sum_{i=1}^{4} c_i w_i \mathbf{p}_i - \mathbf{z} = 0 \\ &\sum_{i=1}^{4} c_i w_i - 1 = 0, \quad 0 < w_{\min} \leq w_i \leq w_{\max} < 1 \end{aligned} \tag{5}$$

where c_i indicates whether the i^{th} foot is in contact with the ground ($c_i = 1$) or not ($c_i = 0$). The variable w_i represents the weighted factors, which should be positive and add up to be 1 with i being the index of all stance legs. The stability margin can be adjusted intuitively by tuning $w_{\min}$ and $w_{\max}$, i.e. the lower and upper bounds. The decision variable $\mathbf{u}$ is thus extended by including the $4N$ weighting factors, yielding

$$\mathbf{u} = [\dddot{\mathbf{c}}_1, \ldots, \dddot{\mathbf{c}}_N, w_{1,1}, \ldots, w_{N,4}] \tag{6}$$

B. CoM Estimator

It is a common situation in load-transportation tasks that the information of the loads such as the mass and loading positions is unknown beforehand. We therefore propose to treat the payloads as a part of the modeling errors and evaluate their approximated effect on walking stability by looking on the resulting CoM shift. That is, the potential impact of the unknown payloads is measured by the deviation of the real CoM from its nominal position that is calculated from the robot model and its body configuration. Since, in general, the payloads will not keep changing after being mounted, it is fair to assume the CoM deviation to be a slowly varying variable whose first-order derivative w.r.t. time is close to zero. Therefore, the CoM deviation can be modeled by augmenting an offset term $\mathbf{c}_e$ in the LIPM (1), which becomes

$$\ddot{\mathbf{c}} = \omega^2(\mathbf{c} + \mathbf{c}_e - \mathbf{z}) \tag{7}$$

We implement a discrete Kalman filter as done in [13], [15] to estimate the propagation of the CoM states and the deviation online

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_{c,k} \\ \hat{\mathbf{y}}_k &= \mathbf{C}\hat{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_{c,k} \end{aligned} \tag{8}$$

TABLE I
ELEMENTS OF THE NOISE COVARIANCES IN THE KALMAN FILTER

	σ_1,σ_2	σ_3,σ_4	σ_5,σ_6
$\mathbf{Q}$	1×10^{-8}	1×10^{-4}	1×10^{-8}
$\mathbf{R}$	1×10^{-6}	1×10^{-2}	-

where $\hat{\mathbf{x}} = \left[\hat{\mathbf{c}}, \dot{\hat{\mathbf{c}}}, \hat{\mathbf{c}}_e\right]$ represents the state of the CoM estimator, $\hat{\mathbf{y}} = \left[\hat{\mathbf{c}}, \ddot{\hat{\mathbf{c}}}\right]$ represents the observation and $\mathbf{u}_c = [u^x_c, u^y_c]$ represents the xy coordinates of the CoP. The process matrices and observation matrices are given as follows

$$\mathbf{A} = \begin{bmatrix} (1 + \frac{1}{2}\omega^2 T_s^2)\mathbf{I}_2 & T_s\mathbf{I}_2 & \frac{1}{2}\omega^2 T_s^2 \mathbf{I}_2 \\ \omega^2 T_s \mathbf{I}_2 & \mathbf{I}_2 & \omega^2 T_s \mathbf{I}_2 \\ 0 & 0 & \omega^2 T_s \mathbf{I}_2 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -\frac{1}{2}\omega^2 T_s^2 \mathbf{I}_2 \\ -\omega^2 T_s \mathbf{I}_2 \\ 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} \mathbf{I}_2 & 0 & 0 \\ \omega^2 \mathbf{I}_2 & 0 & \omega^2 \mathbf{I}_2 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 \\ -\omega^2 \mathbf{I}_2 \end{bmatrix}$$

where $\mathbf{I}_2$ is a 2×2 identity matrix, and T_s is the time interval of discretization that depends on the execution rate of the estimator. Let $\mathbf{Q} \in \mathbb{R}^{6\times6}$ and $\mathbf{R} \in \mathbb{R}^{4\times4}$ denote the process and measurement noise covariances in the Kalman filter, respectively. $\mathbf{Q}$ and $\mathbf{R}$ are both set to be diagonal matrices and their elements are chosen as shown in Table I, where σ_i represents the i^{th} element on the diagonal.

Since there are no force/torque sensors installed at the end of each leg of CENTAURO, contact forces are estimated from joint torque measurements and the robot's dynamics under slow-motion assumptions, through

$$\mathbf{f}_i = (\mathbf{J}^{\mathrm{T}}_{c,i})^+(\mathbf{h} - \mathbf{S}^{\mathrm{T}}\boldsymbol{\tau}), \quad i = 1, \ldots, 4 \tag{9}$$

where $\mathbf{J}_{c,i} \in \mathbb{R}^{3\times(n+6)}$ represents the Jacobian of the i-th foot (n is the number of actuated joints which is 42 for our robot); $(\cdot)^+$ is the pseudo-inverse operation; $\mathbf{S} = [\mathbf{0}_{n\times6} \quad \mathbf{I}_{n\times n}]$ is the selection matrix that picks up the actuated degrees of freedom; $\boldsymbol{\tau}$ is the vector of measured torques in joint space; and $\mathbf{h} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ expresses the nonlinear dynamics caused by centripetal, Coriolis and gravity forces. The position of CoP, $\mathbf{u}_c$, is then calculated by summing up the positions of stance feet weighted by corresponding contact forces.

With regard to the observations, the position of CoM is calculated from measured joint positions and forward kinematics, and, with the assumption that the CoM and the IMU both lie on the trunk, the CoM acceleration can be obtained approximately from the IMU measurements. It also explains why we set a lower belief to the observed IMU accelerations in the Kalman filter, which corresponds to larger measurement noise covariances (i.e., $\sigma_{r,3}$ and $\sigma_{r,4}$ listed in Table I).

The CoM estimator implemented above has been validated in simulations where the ground truth values can be easily obtained. The simulation setup and task descriptions are identical to the *test-2* in Section IV, which will be explained later. The upper figure in Fig. 3 illustrates the real trajectory of the CoM obtained from the simulator and the estimated

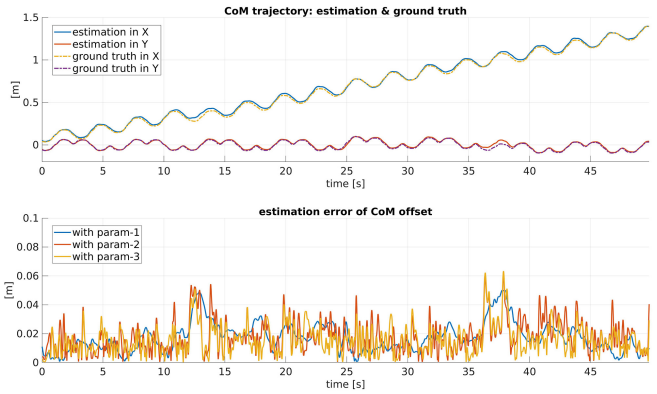

Fig. 3. Validation of the CoM estimator in simulations. Top: Real trajectories and estimated trajectories of the CoM in simulations with $\mathbf{Q}$ and $\mathbf{R}$ listed in Table-I. Bottom: Estimation errors of the CoM offset under different parameters. The blue line corresponds to the case in the upper figure while the red and yellow lines are for the other two sets of parameters.

trajectory by summing up $\hat{\mathbf{c}}$ and $\hat{\mathbf{c}}_e$ from the estimator. The distance between the ground truth CoM and the estimated one, which is the estimation error of the CoM offset, is also shown in the bottom figure of Fig. 3 (the blue line). The errors are often less than 3 cm and peaked at 5 cm only in some instants. To further compare the estimator's performances with different covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, we do simulations with the original parameters shown in Table-I (denoted as param-1) and another two sets of parameters:

- param-2: $\sigma_{q,5} = \sigma_{q,6} = 1 \times 10^{-6}$, others unchanged from param-1;
- param-3: $\sigma_{r,3} = \sigma_{r,4} = 1 \times 10^{-4}$, others unchanged from param-1.

The estimated CoM offset is allowed larger variance (possibly better time responses and smaller delays) under param-2 while the observed joint acceleration is given more trust under param-3. It is evident in Fig. 3 that the new sets of parameters both lead to rapidly varying estimations (and no smaller errors in average), which could undermine the robot's stability when used in compensation, and thus we will adopt the original $\mathbf{Q}$ and $\mathbf{R}$ in the following experiments.

C. Hierarchical Whole-Body Controller

The generation of legged locomotion is a complex task with multiple objectives, such as tracking the reference, regulating the body and distributing the motions while considering various constraints. Because of the flexibility to express multiple objectives in a hierarchical structure with different priorities, numerical optimizations have recently been adopted to tackle the computation of inverse kinematics (IK) and inverse dynamics (ID) [19], [20]. Similarly, we formulate the control of our robot as a hierarchical QP optimization inside *OpenSoT* [21] [1], a light-weight and real-time safe whole-body control framework. To be specific, the whole-body controller with two task hierarchies and equality and inequality constraints are established in the form

$$\begin{aligned} \arg\min_{\dot{\mathbf{q}}_i} & \|\mathbf{J}_i\dot{\mathbf{q}}_i - \dot{\mathbf{x}}_i\|^2 + \lambda\|\dot{\mathbf{q}}_i\|^2 \quad i = 1, 2 \\ \text{s.t.} \quad & \mathbf{A}_{eq,i}\dot{\mathbf{q}}_i = \mathbf{b}_{eq,i} \\ & \underline{\mathbf{c}}_i \leq \mathbf{C}_{in,i}\dot{\mathbf{q}}_i \leq \overline{\mathbf{c}}_i \\ & \mathbf{J}_1\dot{\mathbf{q}}_2 = \mathbf{J}_1\dot{\mathbf{q}}_1^* \end{aligned} \tag{10}$$

where $\dot{\mathbf{x}}_i$ is the task velocity at the ith hierarchy (suppose task $\mathbf{x}_1$ has priority over $\mathbf{x}_2$) and $\mathbf{J}_i$ is the corresponding task Jacobian. $\dot{\mathbf{q}}_1$ is the joint velocity at the primary hierarchy and $\dot{\mathbf{q}}_1^*$ is its optimal solution, whereas $\dot{\mathbf{q}}_2$ is the resultant joint velocity by considering the two hierarchies together. The first two constraints in the optimization above are the user-defined equality and inequality constraints, respectively, and the third constraint represents the requirement of hierarchical division.

Differently from our previous work [12] where all the tasks are organized inside the same hierarchy, in this work we propose to divide the tasks into two hierarchies and attach the top priority to tracking the optimized CoM trajectories. All the tasks and constraints used in our whole-body controller are listed in Table II, together with their corresponding priorities and weighting factors.

TABLE II
TASKS AND CONSTRAINTS IN THE WHOLE-BODY CONTROLLER

Task/Constraint	Priority	Weight
CoM trajectory tracking	first	1.0
pelvis orientation regulation	first	0.1
leg trajectory tracking	second	1.0
arm trajectory tracking	second	1.0
upper body regulation	second	0.1
joint position limit	constraint	-
joint velocity limit	constraint	-

IV. Validation

To validate the proposed reactive locomotion scheme, several trials are conducted to demonstrate that our robot CENTAURO manages to accomplish demanding payload-transportation tasks in different scenarios. The communication between user programs and the hardware is established by *XBotCore*, a real-time (RT) safe robot control framework developed in our lab, which also provides interfaces to non-RT (NRT) processes [22]. All the blocks shown in Fig. 2 are implemented in C++ as NRT ROS nodes and run at the indicated rates on the robot's embedded computer with an Intel i7-6700@3400 Hz processor. Unless otherwise stated, we use the parameters listed in Table III in the following tests.

Three trials of payload-transportation tasks are introduced in this section:

- Test-1: Flat terrain locomotion with payloads;
- Test-2: Flat terrain locomotion with varying payload configurations;
- Test-3: Walking up a step with payloads.

[1]For more about the interfaces, syntax and demonstrations of the whole-body controller *OpenSoT*, readers are also recommended to visit the website of *XBotControl*: *https://github.com/ADVRHumanoids/XBotControl*.

TABLE III

PARAMETERS USED FOR SIMULATIONS AND EXPERIMENTS

Parameter	Value	Parameter	Value
T	0.1 s	$w_{\min}$	0.10
N	20	$w_{\max}$	0.80
α	1×10^{-6}	t_s	1.0 s
β	1.0	t_c	0.5 s
γ	1.0		

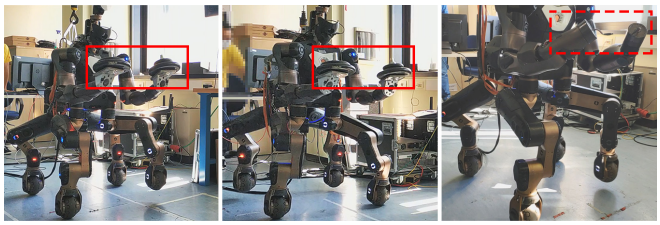

Fig. 4. Snapshots of test-1 where the loaded weights are marked by rectangles with solid red lines. Left: The robot walks with 20 kg payload in the reactive mode. Middle: The robot walks with the payload in the nominal mode. Right: The robot does not carry any payload as a comparison.

In all these tests, the robot carries a total payload of 20 kg distributed equally on the two arms (10 kg on each arm) as marked by the red rectangles in Fig. 4. The 20 kg payload approaches the maximum capacity of the robot's arms, as reported in [2]. Consequently, the CoM of the robot is shifted by a maximum of about 10 cm from its nominal value, depending on the arms configuration. It is worth noting that the robot is unaware of how much and where the payloads are (although they are fixed at the end of the arms for convenience) and whether they are varying during locomotion. To highlight the improvements, the robot is controlled in each test first by using the proposed reactive scheme (called *reactive mode* below) and then by using only nominal CoM information (called *nominal mode*) as a comparison. Readers are recommended to refer to the attached video for more details of these trials.

A. Test-1: Flat terrain locomotion with payloads

Test-1 is the simplest case among all the trials, where the robot walks on flat ground with payloads. We conduct a third test without any payloads to highlight how much the robot is affected by the 20 kg weights, as shown in Fig. 4. By examining the solid and dashed blue lines in Fig. 5, it is clear that the CoM is shifted by about 10 cm along the x-axis but doesn't change much along the y-axis because the payloads are roughly symmetric about the sagittal plane. Fig. 6 displays the estimated contact forces in the reactive-mode experiment. The weight of the whole robot is measured by summing up the z components of the contact forces, as shown in Fig. 7. It reveals that the payloads are indeed about 20 kg by comparing the blue and yellow lines.

We evaluate the walking stability in terms of frequencies and magnitudes of violating the ZMP stability constraints, as displayed in Fig. 8. The reactive mode shows fewer violations as well as a smaller mean (and a much smaller maximum value) when breaking the constraints. Although

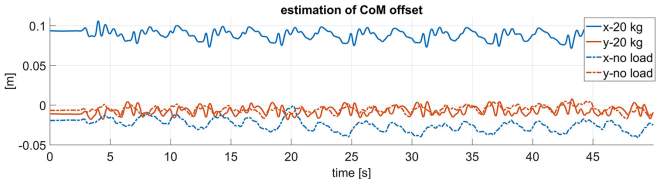

Fig. 5. The estimated CoM offset in test-1. The CoM has been shifted by about 10 cm due to the payloads.

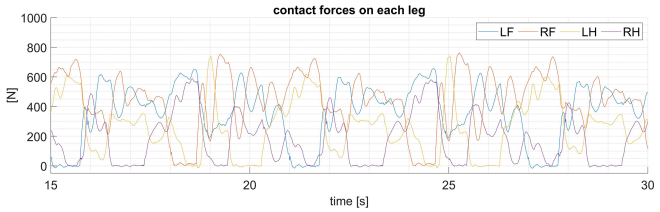

Fig. 6. The estimated contact forces in an interval of 15 s during the reactive locomotion in test-1.

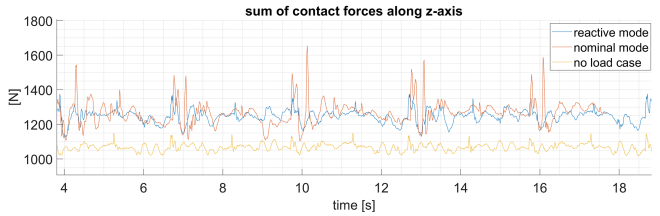

Fig. 7. The sum of the contact forces along z-axis in an interval of 15 s in test-1.

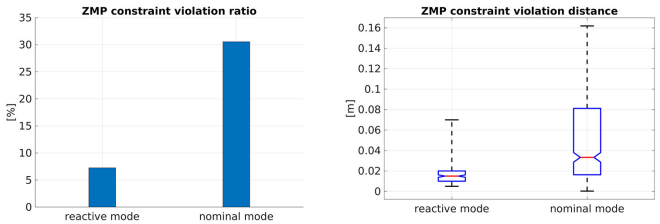

Fig. 8. A straightforward illustration of walking stability in test-1. Left: The proportion of the sampled data where the ZMP constraint is not satisfied. Right: The shortest distance between the ZMP and the corresponding SP for the unstable sample instants.

the motions produced by the MPC planner should always be stable theoretically, tracking of the desired motions is actually not perfect majorly because of the intrinsic compliance of the series elastic actuators (SEAs). In addition, the walking stability can be explained by the attached video, where the robot is shown to lose its balance and bump into the ground heavily during the nominal-mode locomotion. The corresponding contact forces, presented by the red line in Fig. 7, also demonstrate larger peak values due to the heavy collisions. To prevent damaging the robot due to high impacts we terminated the experiment of nominal-mode locomotion only after about 15 s. In order not to damage to the hardware, we conduct only the reactive-mode tests on the robot whereas the following nominal-mode tests are in simulations.

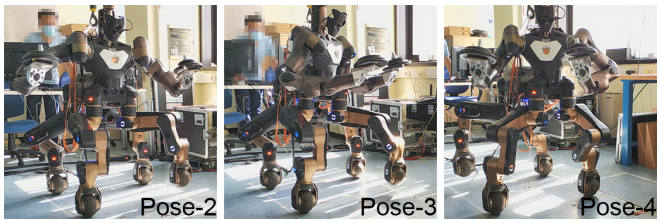

Fig. 9. In test-2, the arm configurations are changed online to produce varying impacts from the payloads.

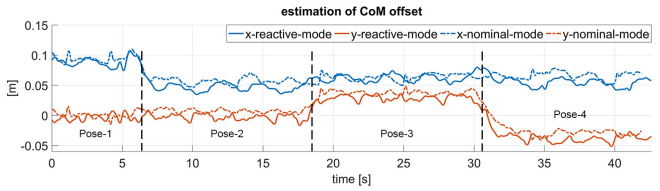

Fig. 10. Estimations of the CoM offset in test-2. The CoM is affected differently under distinct payload configurations.

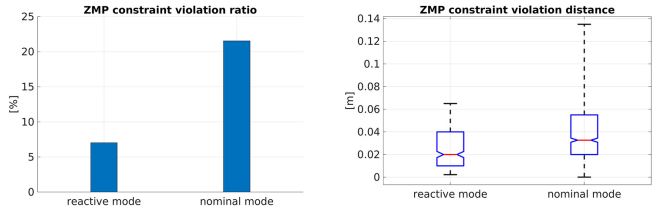

Fig. 11. An illustration of walking stability in test-2.

B. Test-2: Flat terrain locomotion with varying payload configurations

In test-2, the payloads are varying on the fly by modulating the postures of the arms (without informing the locomotion controller), and the robot is expected to adapt its movements accordingly. The robot starts from the initial pose (denoted as *pose-1*) and sequentially change to three other poses (denoted as *pose-2*, *pose-3* and *pose-4*, respectively) during walking, as displayed in Fig 9.

Fig. 10 displays the estimated CoM offset, which reveals that the CoM is affected differently under distinct payload configurations. The experimental results demonstrate that the robot manages to adapt to different payload configurations seamlessly in the reactive mode. However, the nominal mode produces stable locomotion only when the payloads are close enough to the trunk, i.e., at *pose-2*. We further evaluate the walking stability by calculating how much the ZMP constraint is violated, as illustrated in Fig. 11.

C. Test-3: Walking up a step with payloads

In test-3, the robot is controlled to walk up a step of 15 cm with the payload. The robot manages to walk up the step in the reactive mode, whereas it always fails in the nominal mode as shown in the video. Therefore, we only demonstrate the walking stability of the reactive locomotion in Fig. 12. This trial, consequently, better validates the traversing capacity of CENTAURO and the proposed reactive locomotion generation scheme.

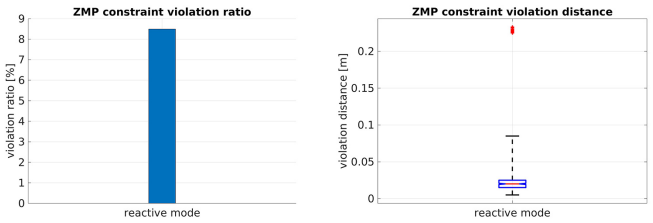

Fig. 12. An illustration of walking stability in test-3 where the outliers are marked by red crosses.

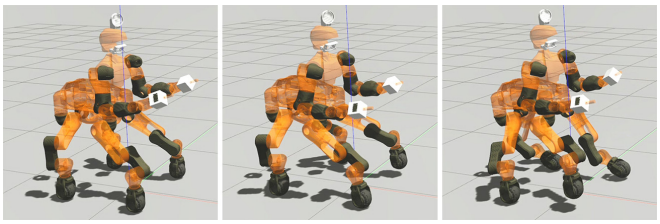

Fig. 13. The robot fails in walking when γ is set to be 0.

D. Test-4: Evaluation of the γ-related term

We have introduced a γ-related term in the optimization (4) to trade off large stability margins and equally stretched legs. To demonstrate the necessity of having this trade-off term for the reactive locomotion, another test is carried out in simulations with the parameter γ set to be 0, which means that the motion planner will only focus on maximizing the stability margins. As displayed in Fig. 13, the pelvis moves far too back to compensate for payloads while pushing the front legs close to the kinematic limits. The controller fails to generate stable motions after only several seconds and then the robot loses its balance.

V. Conclusion

In this work, we proposed a novel planning and control framework that implements adaptive legged locomotion under unknown and online-varying heavy payloads carried by the quadruped robot CENTAURO.

The main components of the proposed framework are i) an MPC-based motion planner, ii) a QP-based whole-body controller and iii) a CoM estimator that evaluates the influence of the payloads on the robot. A cost term that balances the pursuit of maximizing stability margins and evenly stretching the legs is also introduced, enabling the generation of feasible motions during the reactive-mode locomotion. Experiments and simulations have been conducted to validate that, by using the proposed reactive walking strategy, the robot is able to accomplish demanding load-transportation tasks with significantly heavy and online-varying payloads in different scenarios with improved walking stability.

One of the possible future directions is to explore impedance control and further improve the interaction behaviors through variable stiffness control of the joints.

REFERENCES

[1] T. Klamt, M. Kamedula, H. Karaoguz, N. Kashiri, A. Laurenzi, C. Lenz, D. Leonardis, E. Mingo Hoffman, L. Muratore, D. Pavlichenko, F. Porcini, D. Rodriguez, Z. Ren, F. Schilling, M. Schwarz, M. Solazzi, M. Felsberg, A. Frisoli, M. Gustmann, P. Jensfelt, K. Nordberg, J. Rossmann, L. Baccelliere, U. Suess, N. G. Tsagarakis, S. Behnke, X. Chen, D. Chiaradia, T. Cichon, M. Gabardi, P. Guria, and K. Holmquist, "Flexible disaster response of tomorrow: Final presentation and evaluation of the centauro system," *IEEE Robotics and Automation Magazine*, vol. 26, pp. 59–72, 2019.

[2] N. Kashiri, L. Baccelliere, L. Muratore, A. Laurenzi, Z. Ren, E. M. Hoffman, M. Kamedula, G. F. Rigano, J. Malzahn, S. Cordasco, and Others, "CENTAURO: A Hybrid Locomotion and High Power Resilient Manipulation Platform," *IEEE Robotics and Automation Letters (RAL)*, vol. 4, pp. 1595–1602, 2019.

[3] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin *et al.*, "Chimp, the cmu highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, pp. 209–228, 2015.

[4] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke, "Hybrid driving-stepping locomotion with the wheeled-legged robot momaro," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5589–5595.

[5] Y. Wu, P. Balatti, M. Lorenzini, F. Zhao, W. Kim, and A. Ajoudani, "A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant," *IEEE Robotics and Automation Letters (RAL)*, vol. 4, pp. 3593–3600, 2019.

[6] 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 (ICRA)*, 2003, pp. 1620–1626.

[7] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.

[8] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast Trajectory Optimization for Legged Robots Using Vertex-Based ZMP Constraints," *IEEE Robotics and Automation Letters (RAL)*, vol. 2, pp. 2201–2208, 2017.

[9] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, pp. 719–737, 2010.

[10] P. B. Wieber, "Viability and predictive control for safe locomotion," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1103–1108.

[11] A. Laurenzi, E. M. Hoffman, and N. G. Tsagarakis, "Quadrupedal walking motion and footstep placement through Linear Model Predictive Control," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2267–2273.

[12] S. Jingyuan, Y. Yangwei, Z. Xuran, A. Albertus, and C. M. Chew, "Towards more possibilities: Motion planning and control for hybrid locomotion of wheeled-legged robots," *IEEE Robotics and Automation Letters (RAL)*, vol. 5, pp. 3723–3730, 2020.

[13] B. J. Stephens, "State estimation for force-controlled humanoid balance using simple models in the presence of modeling error," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3994–3999.

[14] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013.

[15] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 67–73.

[16] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. G. Caldwell, and C. Semini, "Online payload identification for quadruped robots," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4889–4896.

[17] R. Orsolino, M. Focchi, S. Caron, G. Raiola, V. Barasuol, and C. Semini, "Feasible Region: an Actuation-Aware Extension of the Support Region," 2019.

[18] C. Gehring, C. D. Bellicoso, P. Fankhauser, S. Coros, and M. Hutter, "Quadrupedal locomotion using trajectory optimization and hierarchical whole body control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4788–4794.

[19] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research (IJRR)*, vol. 33, pp. 1006–1028, 2014.

[20] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, "Keep rollin'—whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robotics and Automation Letters (RAL)*, vol. 4, pp. 2116–2123, 2019.

[21] E. M. Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: Insights and examples using opensot," in *IEEE International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 736–741.

[22] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *IEEE International Conference on Robotic Computing (IRC)*, 2017, pp. 77–80.