Whole-body motion tracking for a quadruped-on-wheel robot via a compact-form controller with improved prioritized optimization

Wenqian Du, Mohamed Fnadi, Faïz Ben Amar

Whole-body Motion Tracking for a Quadruped-on-wheel Robot via a Compact-form Controller with Improved Prioritized Optimization

Wenqian Du[1], Mohamed Fnadi[1] and Faïz Benamar[1]

Abstract—This paper develops a more general dynamics controller to generate whole-body behaviors for a quadruped-on-wheel robot. To track the quadruped centroidal motion, the wheeled motion is achieved by combining the wheel contact constraints and the centroidal momentum/dynamics model. The dynamics controller is based on a new hybrid hierarchical and prioritized weighted optimization framework. We propose one concept of a recursively updated dynamics model and this model enables to integrate the new prioritized weighted scheme in the hierarchical framework. In contrast with the conventional weighted scheme, we propose to use null-space projections among its sub-tasks. Then the prioritized impedance controller is proposed and integrated in our dynamics model, which enables to influence the hierarchical and prioritized weighted tasks in a decoupled way. The task accelerations in the two schemes are extracted with quadratic forms depending on the actuated torque and the prioritized impedance force using null-space based inverse dynamics. The inequality constraints are modified to ensure the compatibility with the hybrid convex optimization. This dynamics controller is more general and its algorithm is given completely which enables our robot to track the centroidal motion on rough terrain and handle other missions in three simulation scenarios.

Index Terms—Optimization and optimal control, dynamics, legged robots, wheeled robots.

I. INTRODUCTION

FROM a practical point of view, quadruped-on-wheel robots combine the advantages of conventional vehicles and legged robots, which have more versatile locomotion modes from rolling to walking, climbing, running and jumping, etc. This kind of robot is not tied to discontinuous natural environments and also reduce energy consumption on continuous terrain using only wheeled motion. Therefore, it has more potential for universe exploration, mining, industry or disaster rescue. However, these enhanced locomotion skills come with more complex motion and torque control.

In this state of the art, only quadruped-on-wheel robots with active redundant actuators are reviewed. The Hylos robot [1] is developed in ISIR laboratory and is capable to run on rough terrain by changing its base center-of-mass (CoM) position and optimizing the joint torques and contact forces using
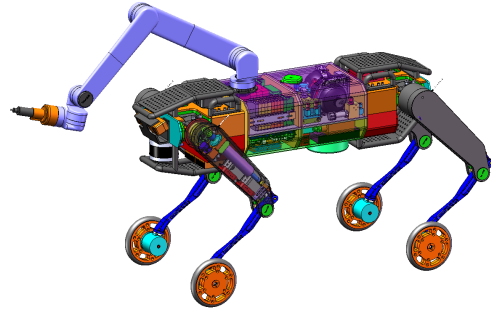
Fig. 1. "Towr*ISIR*", a quadruped-on-wheel robot developed in ISIR lab is equipped with 12 hydraulic actuators and 4 wheel motors. Each leg has 4 joints, including hip abduction/adduction joint (HAA), hip flexion/extension joint (HFE), knee flexion/extension joint (KFE) and wheel forward/backward joint (WFB). One manipulator is installed on its base.

structure redundancy. The Complios robot [2] is designed with active and passive compliant joints driven by Series-Elastic Actuation which can climb obstacles using reinforcement learning. Authors in [3] developed a hydraulic actuated legged-on-wheel robot, which has a similar control method as Hylos that computes the inverse kinematics to track the desired base trajectory and optimize the posture. However, the simplified dynamics model or the kinematics-based controller limits the diversity of dynamic behaviors. Another quadruped-on-wheel robot is developed in ETH Zurich, ANYmal [4] which combines the advantages of the legged motion and wheeled motion, in which contact points are estimated, and hierarchical optimization is applied to track the contact footholds which come from a model based predictive controller. However, most of the works track the base motion without considering the whole-robot CoM motion. As we know, the system CoM occupies a special place in its dynamics. The system translational and angular momentum can be defined here naturally and authors in [5] verify that control of the centroidal angular momentum can enable more natural rich whole-body motion. Recently, one centaur-like robot, CENTAURO [6] is developed to track the system CoM motion by using only the first-order inverse kinematics. However, there is no application for tracking the centroidal motion of quadruped-on-wheel robots using whole-robot dynamics.

For whole-body torque control, compared with the pseudo-inverse based dynamics method in [7], optimization based techniques can deal with general constraints. The optimization solver is based on Quadratic Programming (QP), in weighted or hierarchical scheme. Weighted schemes optimize all tasks

in one QP [8], unfeasible behaviors may occur when weights are not well distributed or the robot degrees of freedom (DOF) are not enough and all tasks are fulfilled with a compromise. Furthermore, various feedback controllers are integrated which may influence all tasks together and cannot be accepted. The hierarchical cascade QP framework (HQP) is proposed in [9] which ensures strict hierarchies so that low priority tasks cannot conflict with high level tasks. Moreover, [10] generates several types of motions (sitting, yoga, walking) by applying hierarchical QPs and Stack of Tasks that enable adding or removing tasks at any moment. Furthermore, two kinds of hierarchical least square optimization algorithms are developed in [11] and [12], in which high-priority tasks are solved initially and the expected task errors serve as equality constraints for low-priority tasks. The hierarchical conic optimization [11] is applied to control the centroidal angular momentum to generate humanoid behaviors. Pseudo inverses are integrated in HQP to optimize hybrid tasks related to motion and force for a quadruped robot [12]. In addition, [13] proposes to avoid equality constraints which are generated between hierarchies, by using null-space projectors in the cost function and inequality constraints. The weighted and hierarchical QPs are combined together in [14]. In these works, for each task, a specific feedback impedance controller is added to enable the compliance of a specific task. However, these works did not handle the consistency of general impedance controllers with task hierarchies, especially the ones which indirectly influence the whole-body motions.

In this paper, given the quadruped-CoM translational motion reference, the wheeled motion is firstly generated by using the wheel contact constraints and the decomposed centroidal momentum and dynamics models. Then to track the whole-body motion references, we develop one compact-form torque controller which has several sub-contributions: This controller is based on one hybrid hierarchical-weighted QP solver which is achieved by a proposed recursively updated dynamics model. This model is used to extract the task accelerations in each hierarchy with quadratic forms. For one hierarchy with several sub-tasks, in contrast with the conventional weighted QP, we firstly propose that priorities exist in the weighted scheme, and the proposed model enables to integrate this new weighted scheme into the hierarchical control framework. Then we propose to combine our prioritized impedance controllers into the task extractions by modifying the recursively updated dynamics model. By using our method, the designed hierarchical tasks and weighted sub-tasks are influenced by the impedance forces in a decoupled way. Furthermore, this method can handle various impedance controllers, especially for the general ones which indirectly influence the designed tasks. With these fundamental theories, the new hybrid hierarchical and prioritized weighted control scheme is more general and complete compared with existing works. The whole algorithm is completely described and can be used for other high-DOF locomotion robots.

The remainder of this paper is organized: In Section II, system modeling is firstly detailed. Then the compact form dynamics controller is developed in Section III. In Section IV, we test our method through numerical simulations. Section V
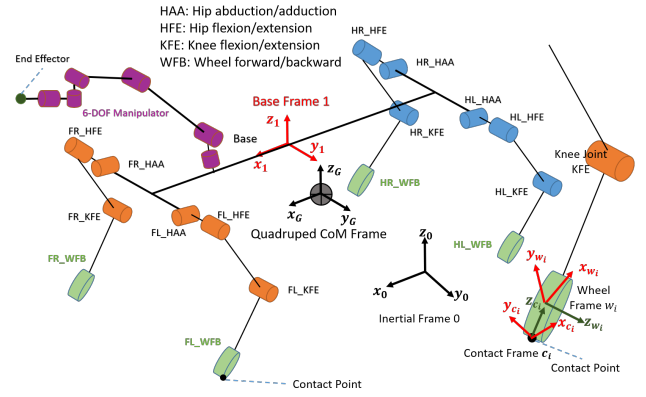

Fig. 2. Connectivity of Towr*ISIR*: The robot has $N_B = 23$ bodies, the manipulator accounts for 6 bodies and the quadruped part has 17 bodies. The quadruped-CoM frame G is set parallel with the base frame 1. The frames relating to the inverse kinematics are zoomed in, including the contact frame c_i, wheel frame w_i. F, L, R, H represent front, left, right and hind.

closes the paper with conclusion.

II. SYSTEM MODELING

"Towr*ISIR*", a quadruped-on-wheel robot is designed in ISIR laboratory (see Fig. 1). It is equipped with one floating base and four branches, each branch has three active-actuated leg joints and one wheel joint. The extra wheel joints induce more complexity to the kinematics/dynamics model. To track the quadruped-CoM motion, the real-time wheel motion is generated by combining wheel contact constraints and the centroidal momentum/dynamics model.

A. Whole-Robot Dynamics Modeling

This robot has $n = 28$ DOF in which 6 DOF is relative to the floating base motion (see Fig. 2). The robot configuration is represented as

$$\boldsymbol{q} = \begin{bmatrix} \mathbf{x}_b^T & \boldsymbol{q}_r^T & \boldsymbol{q}_w^T & \boldsymbol{q}_m^T \end{bmatrix}^T \in \mathbb{R}^{n_b+n_r+n_w+n_m=n}, \quad (1)$$

where $\mathbf{x}_b = [\boldsymbol{\theta}_b^T, \boldsymbol{p}_b^T]^T$ is the orientation and position of the floating base in inertial global coordinate, $\boldsymbol{q}_r \in \mathbb{R}^{n_r}$ represents $n_r = 12$ revolute joint configurations, $\boldsymbol{q}_w \in \mathbb{R}^{n_w}$ represents $n_w = 4$ continuous wheel joint configurations, $\boldsymbol{q}_m \in \mathbb{R}^{n_m}$ represents $n_m = 6$ revolute arm joint configurations. We define $\boldsymbol{q}_g = \begin{bmatrix} \boldsymbol{q}_r^T & \boldsymbol{q}_w^T & \boldsymbol{q}_m^T \end{bmatrix}^T$ to represent the actuated joints.

The generalized dynamics model in joint space is derived by Newton-Euler using spatial vectors [15] as follows,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{J}_c^T \boldsymbol{F}_c + \boldsymbol{S}^T \boldsymbol{\tau} = \tilde{\boldsymbol{\tau}}, \quad (2)$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, $\boldsymbol{C} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^n$, $\boldsymbol{F}_c \in \mathbb{R}^{3n_c}$ are the inertia matrix , centripetal & coriolis force, gravity force and contact forces respectively. n_c is the contact point number. $\boldsymbol{J}_c \in \mathbb{R}^{3n_c \times n}$ is the contact Jacobian, only considering the translational motion of all contact points. $\boldsymbol{\tau} \in \mathbb{R}^{n_r+n_w}$ is the actuated torque and $\boldsymbol{S} \in \mathbb{R}^{(n_r+n_w)\times n}$ is the selection matrix.

In this paper, we focus on the rolling motion and all wheels are assumed to be always in contact with the ground. Then the constant-dimension contact Jacobian is derived as below,

$$\boldsymbol{J}_c = \begin{bmatrix} \boldsymbol{J}_c^{\omega_b} & \boldsymbol{J}_c^{v_b} & \boldsymbol{J}_c^r & \boldsymbol{J}_c^w & \boldsymbol{J}_c^m \end{bmatrix}, \quad (3)$$

where $\boldsymbol{J}_c^{\omega_b}$ and $\boldsymbol{J}_c^{v_b}$ relate to the base angular and translational motion respectively, which are both $\in \mathbb{R}^{n_c \times 3}$. $\boldsymbol{J}_c^r \in \mathbb{R}^{n_c \times n_r}$ and $\boldsymbol{J}_c^w \in \mathbb{R}^{n_c \times n_w}$ relate to the revolute and wheel joint motion respectively. $\boldsymbol{J}_c^m \in \mathbb{R}^{n_c \times n_m} = \boldsymbol{0}$ relates to the manipulator joints. Assuming that there is no slippage between the contact points and the ground, the velocity of the wheel contact points is expressed by the following constraint (4a) and its time derivative is in (4b),

$$\boldsymbol{v}_c = \boldsymbol{J}_c \dot{\boldsymbol{q}} = \boldsymbol{J}_c^{\omega_b} \boldsymbol{\omega}_b + \boldsymbol{J}_c^{v_b} \boldsymbol{v}_b + \boldsymbol{J}_c^r \dot{\boldsymbol{q}}_r + \boldsymbol{J}_c^w \dot{\boldsymbol{q}}_w = \boldsymbol{0}, \quad (4a)$$

$$\dot{\boldsymbol{v}}_c = \boldsymbol{J}_c^{\omega_b} \dot{\boldsymbol{\omega}}_b + \boldsymbol{J}_c^{v_b} \dot{\boldsymbol{v}}_b + \boldsymbol{J}_c^r \ddot{\boldsymbol{q}}_r + \boldsymbol{J}_c^w \ddot{\boldsymbol{q}}_w + \dot{\boldsymbol{J}}_c \dot{\boldsymbol{q}} = \boldsymbol{0}, \quad (4b)$$

where $\boldsymbol{v}_c$ is the velocity of all wheel contact points. $\boldsymbol{\omega}_b$ and $\boldsymbol{v}_b$ denote the base angular and translational velocity.

B. Consistent Rolling Motion Generation

We set a frame at the quadruped CoM to be parallel with the inertial frame, which enables convenience to control the quadruped-CoM motion in the inertial frame. In the control framework shown in Fig. 4, we concentrate on the rolling mode and the quadruped-CoM translational motion serves as the reference. Its forward motion is achieved by the wheeled locomotion and its vertical and lateral motion are handled by the legged suspension, e.g. squirting motion. Its desired leg-joint references $(\boldsymbol{q}_r^d, \dot{\boldsymbol{q}}_r^d, \ddot{\boldsymbol{q}}_r^d)$ are generated by one legged motion generator which also outputs the desired base angular motion, shown in Fig. 4, however, it exceeds the topic of this paper. The desired legged motion serves as the inputs of the wheel motion generator. Since the wheel contact points always change even when the robot squirts down, to track the quadruped-CoM motion, the wheel motion should be consistent with the legged suspension.

The wheel motion generator is developed by combining the wheel contact constraints and the quadruped centroidal momentum/dynamics model. The inputs of the wheel velocity and acceleration generators are respectively $\{\boldsymbol{p}_G, \boldsymbol{v}_G^d, \boldsymbol{\theta}_b, \boldsymbol{\omega}_b^d, \boldsymbol{q}_r, \dot{\boldsymbol{q}}_r^d, \boldsymbol{q}_w\}$ and $\{\boldsymbol{p}_G, \boldsymbol{v}_G, \dot{\boldsymbol{v}}_G^d, \boldsymbol{\theta}_b, \boldsymbol{\omega}_b, \dot{\boldsymbol{\omega}}_b^d, \boldsymbol{q}_r, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r^d, \boldsymbol{q}_w, \dot{\boldsymbol{q}}_w\}$, where $\boldsymbol{p}_G$ and $\boldsymbol{v}_G$ denote the quadruped-CoM position and translational velocity, respectively. The global solving process: the base translational velocity $\boldsymbol{v}_b$ and acceleration $\dot{\boldsymbol{v}}_b$ can be extracted out from the decomposed centroidal momentum and dynamics equations respectively, and $\boldsymbol{v}_b$ and $\dot{\boldsymbol{v}}_b$ are substituted into the inverse kinematics equations in (4a) and (4b). Then the wheel velocity $\dot{\boldsymbol{q}}_w$ and acceleration $\ddot{\boldsymbol{q}}_w$ are derived using Moore-Penrose inverses.

1) Wheel Velocity Generator: The wheel motion $\dot{\boldsymbol{q}}_w$ can be derived by combining one kinematic velocity constraint in (4a), and the quadruped centroidal momentum model which is divided into parts relating to the base motion, legged motion and the wheel motion as follows,

$$\mathbf{I}_G \begin{bmatrix} \boldsymbol{\omega}_G \\ \boldsymbol{v}_G \end{bmatrix} = {}^1\mathbf{X}_G^T \mathbf{I}_{quad} \begin{bmatrix} \boldsymbol{\omega}_b \\ \boldsymbol{v}_b \end{bmatrix} + {}^1\mathbf{X}_G^T \sum {}^j\mathbf{X}_1^T \mathbf{I}_j {}^b\mathbf{v}_j + \mathbf{Q}_w \dot{\boldsymbol{q}}_w, \quad (5)$$

where $j \in$ **Leg Bodies** which include bodies except for the base, wheels and the manipulator. $\mathbf{I}_{quad}$ is the quadruped composite inertia in the base frame. $\boldsymbol{\omega}_G$ is the quadruped-CoM angular velocity. ${}^b\mathbf{v}_j$ is the spatial velocity of leg body j relative to the base. ${}^1\mathbf{X}_G$ is the transformation matrix from

---

Algorithm 1: Derivations of $\mathbf{Q}_r \ddot{\boldsymbol{q}}_r$ and $\mathbf{Q}_{bias}$.

Define $\mathbf{B}$, $\mathbf{E}$, $\mathbf{e}_j$ and $\mathbf{b}_j$ as middle-process variables.
$\mathbf{B} = \mathbf{v}_1 \times \mathbf{I}_1 \mathbf{v}_1$, $\mathbf{E} = \mathbf{0}$, $\mathbf{e}_1 = \mathbf{0}$, $\ddot{\boldsymbol{q}}_w = \mathbf{0}$, $\mathbf{b}_1 = \mathbf{0}$.
for $j = 2$ *to* 17 **do**
  $\quad \mathbf{v}_j = {}^j\mathbf{X}_{p(j)} \mathbf{v}_{p(j)} + \boldsymbol{\Phi}_j \dot{\boldsymbol{q}}_j$
  $\quad \mathbf{e}_j = {}^j\mathbf{X}_{p(j)} \mathbf{e}_{p(j)} + \boldsymbol{\Phi}_j \ddot{\boldsymbol{q}}_j$
  $\quad \mathbf{E} = \mathbf{E} + {}^j\mathbf{X}_1^T \mathbf{I}_j \mathbf{e}_j$
  $\quad \mathbf{b}_j = {}^j\mathbf{X}_{p(j)} \mathbf{b}_{p(j)} + \mathbf{v}_j \times \boldsymbol{\Phi}_j \dot{\boldsymbol{q}}_j$
  $\quad \mathbf{B} = \mathbf{B} + {}^j\mathbf{X}_1^T (\mathbf{I}_j \mathbf{b}_j + \mathbf{v}_j \times \mathbf{I}_j \mathbf{v}_j)$
end

$\mathbf{Q}_r \ddot{\boldsymbol{q}}_r = {}^1\mathbf{X}_G^T \mathbf{E}$ and $\mathbf{Q}_{bias} = \begin{bmatrix} \boldsymbol{\omega}_G \\ \boldsymbol{v}_G \end{bmatrix} \times \mathbf{I}_G \begin{bmatrix} \boldsymbol{\omega}_G \\ \boldsymbol{v}_G \end{bmatrix} - {}^1\mathbf{X}_G^T \mathbf{B}$.

---

quadruped CoM frame to the base frame, shown in Fig. 2. ${}^j\mathbf{X}_1$ transforms vectors from base frame to body frame j. $\mathbf{I}_j$ represents the body j inertia. $\mathbf{Q}_w \in \mathbb{R}^{6 \times 4}$ is derived as,

$$\mathbf{Q}_w = {}^1\mathbf{X}_G^T \begin{bmatrix} {}^{w_1}\mathbf{X}_1^T \mathbf{I}_{w_1} \boldsymbol{\Phi}_{w_1} & \cdots & {}^{w_4}\mathbf{X}_1^T \mathbf{I}_{w_4} \boldsymbol{\Phi}_{w_4} \end{bmatrix}, \quad (6)$$

where $\mathbf{I}_{w_k}$ and $\boldsymbol{\Phi}_{w_k}$ $(k \in \{1, \cdots, 4\})$ denotes the inertia and free modes of the wheel k respectively (see Fig. 2). ${}^{w_k}\mathbf{X}_1$ transforms spatial vectors from the base frame to the wheel frame. From (5), $\boldsymbol{v}_b$ can be extracted and substituted to (4a), then $\dot{\boldsymbol{q}}_w$ can be derived using Moore–Penrose inverse.

2) Wheel Acceleration Generator: The wheel acceleration $\ddot{\boldsymbol{q}}_w$ is derived using the kinematic acceleration constraint in (4b), and the centroidal dynamics constraint which is also divided by the base motion, legged motion, wheeled motion and one bias item as below,

$$\mathbf{I}_G \begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \dot{\boldsymbol{v}}_G \end{bmatrix} = {}^1\mathbf{X}_G^T \mathbf{I}_{quad} \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\boldsymbol{v}}_b \end{bmatrix} + \mathbf{Q}_r \ddot{\boldsymbol{q}}_r + \mathbf{Q}_w \ddot{\boldsymbol{q}}_w - \mathbf{Q}_{bias}, \quad (7)$$

where $\mathbf{Q}_r \ddot{\boldsymbol{q}}_r$ and $\mathbf{Q}_{bias}$ are achieved in Algorithm 1 using the indexes shown in Fig. 3. In the algorithm, $\mathbf{v}_j$ represents the leg body j spatial velocity, and $p(j)$ represents the parent body of body j. From (7), $\dot{\boldsymbol{v}}_b$ can be extracted out and substituted to (4b), then $\ddot{\boldsymbol{q}}_w$ can be derived. The detailed mathematical derivations for achieving the wheeled motion are omitted here. The by-products contain the quadruped CoM angular motion and the base translational motion.
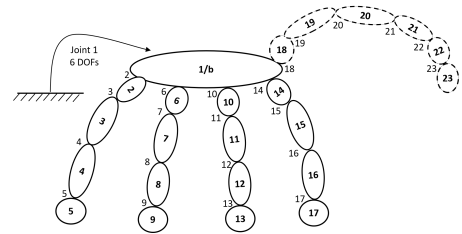
Fig. 3. Indexes for Algorithm 1. Numbers inside and outside each circle represent the body index and joint index respectively. Indexes from 1 to 17 represent the Quadruped Bodies. Dashed circles represent the manipulator.

C. Quadruped Centroidal Jacobian

To control the CoM motion of the **Quadruped Bodies** which include bodies except the manipulator, we derive the Jacobian at the quadruped CoM as below,

$$\boldsymbol{J}_G = [{}^1\mathbf{X}_G^T (\boldsymbol{U} \boldsymbol{M}_{quad} \boldsymbol{U}^T) {}^1\mathbf{X}_G]^{-1} ({}^1\mathbf{X}_G^T \boldsymbol{U} \boldsymbol{M}_{quad}), \quad (8)$$

$$\boldsymbol{M}_{quad} = \sum \boldsymbol{J}_i^T \mathbf{I}_i \boldsymbol{J}_i, \ i \in \text{Quadruped Bodies} \quad (9)$$

where $\boldsymbol{U} = [\mathbf{1}_{6\times 6}, \mathbf{0}_{6\times(n-6)}]$. It is noticed that $\boldsymbol{M}_{quad} \in \mathbb{R}^{n\times n}$ is the quadruped mass matrix, which means that components in $\boldsymbol{M}_{quad}$ relating to the manipulator bodies are equal to zero. And $\boldsymbol{M}_{quad}$ is one by-product when getting the whole-body generalized mass matrix $\boldsymbol{M}$ through the Composite Rigid Body algorithm [17]. The quadruped composite inertia in the base frame is $\mathbf{I}_{quad} = \boldsymbol{U}\boldsymbol{M}_{quad}\boldsymbol{U}^T$ and the composite rigid body inertia at the quadruped CoM is $\mathbf{I}_G = {}^1\mathbf{X}_G^T \mathbf{I}_{quad} {}^1\mathbf{X}_G$. With $\boldsymbol{J}_G$, tasks relating to quadruped CoM can be integrated into the prioritized operational space control framework.

Apart from the quadruped-CoM motion and wheel motion, the manipulator end-effector motion is generated using the Kinematics-Dynamics Library (KDL) [19]. It is noticed that conventional acceleration at the end effector should be modified to be consistent with spatial accelerations.

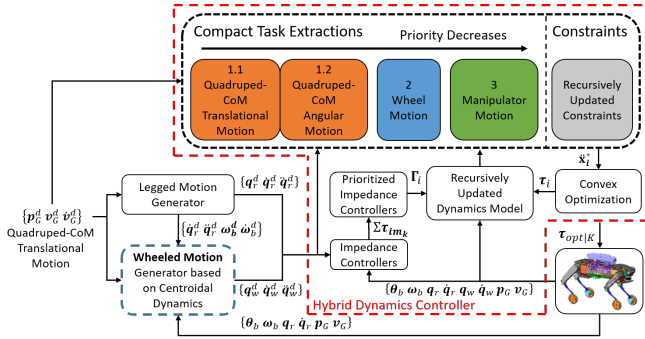Fig. 4. Control Framework: Given the quadruped-CoM translational motion reference, the wheel motion is achieved, detailed in Section II-B. The motion references serve as the inputs of our dynamics controller in the dashed red box, including the proposed recursively updated dynamics model, task extractions, prioritized impedance controllers, recursively updated inequality constraints and convex optimization. Three hierarchies are defined and each color stands for one hierarchy. The highest hierarchy includes the quadruped-CoM translational/angular motion sub-tasks. The controller is detailed in Section III. The variables in this framework are detailed throughout this paper.

III. COMPACT FORM DYNAMICS CONTROLLER

In this section, we develop a compact-form dynamics controller to track the whole-robot motion. The control framework is shown in Fig. 4. Initially, we propose one concept based on one recursively updated dynamics model. For hierarchies with only one task, we extract the task accelerations from this model using the prioritized inverse dynamics method. In addition, for hierarchies with more than one task, our proposed model enables to integrate one new prioritized weighted scheme into the hierarchical scheme, and the sub-task quadratic-form extractions are achieved depending on the actuated torque for the first time. To achieve multi-task compliance, various impedance controllers are proposed to be integrated in our proposed model to ensure consistency with the task hierarchies. Then the inequality constraints are modified recursively to be compatible with our approach. Finally, the compact algorithm is given completely.

A. Recursive Updated Dynamics Model for Strict Priorities

According to the prioritized operational-space inverse dynamics, the following torque equation embodies a multi-

task control hierarchy (with K level tasks),

$$\tilde{\boldsymbol{\tau}} = \boldsymbol{J}_c^T \boldsymbol{F}_c + \sum_{i=1}^{K} \boldsymbol{N}_{pre(i)}^T \boldsymbol{S}^T \boldsymbol{\tau}_i, \tag{10}$$

where $\boldsymbol{N}_{pre(i)}$ is the null-space projector and can be calculated recursively as $\boldsymbol{N}_{pre(i+1)} = \boldsymbol{N}_{pre(i)} - \bar{\boldsymbol{J}}_{i|pre(i)} \boldsymbol{J}_{i|pre(i)}$. $\boldsymbol{J}_{i|pre(i)} = \boldsymbol{J}_i \boldsymbol{N}_{pre(i)}$ denotes the prioritized operational space Jacobian for task i, which is in the null space of previous tasks, represented as $pre(i)$. $\boldsymbol{N}_{pre(1)} = \boldsymbol{N}_c$ represents the null-space projector of the contact Jacobian. $\bar{\boldsymbol{J}}_{i|pre(i)}^T$ is the prioritized contact dynamics-consistent inverse of Jacobian. In this way, the generated torque for each level, $\boldsymbol{\tau}_i$, will not conflict the achievement of tasks higher than level i.

For K hierarchies, the desired task acceleration references in the operational space is $\ddot{\mathbf{x}}_1^*$ to $\ddot{\mathbf{x}}_K^*$. The error between the desired reference $\ddot{\mathbf{x}}_i^*$ and the expected $\ddot{\mathbf{x}}_i$ can be used as inputs of the optimization solver. In contrast with the conventional prioritized operational space control which uses one dynamics model, we propose to modify the dynamics model in (2) in a recursive way. To be general, for hierarchy i, the dynamics model is updated in an additional recursive way by combining all tasks in hierarchy $< i$.

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C} + \boldsymbol{G} = \boldsymbol{J}_c^T \boldsymbol{F}_c + \sum_{t=1}^{i-1} \boldsymbol{N}_{pre(t)}^T \boldsymbol{S}^T \boldsymbol{\tau}_t + \boldsymbol{S}^T \boldsymbol{\tau}_i, \tag{11}$$

when there are feedback compliance controllers, this model should be modified to (29) to combine our proposed prioritized impedance controllers in Section III-D. It also enables to integrate our prioritized weighted scheme in Section III-C into the hierarchical control scheme in Section III-B. By using this model, we achieve each quadratic-form extraction of the tasks in the two schemes in a dynamically recursive way, which avoids equality constraints.

B. Task Extraction for the Hierarchical Framework

With the updated dynamics model in (11), we can achieve the task i acceleration and extract its quadratic form depending on the actuated torque as below,

$$\ddot{\mathbf{x}}_i = \boldsymbol{A}_i^T \boldsymbol{\tau} + \boldsymbol{B}_i, \tag{12}$$

where the general matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are derived by introducing $\boldsymbol{h}_{i|pre(i)} = \boldsymbol{\mu}_{i|pre(i)} + \boldsymbol{\rho}_{i|pre(i)}$ as follows,

$$\boldsymbol{A}_i^T = \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \bar{\boldsymbol{J}}_{i|pre(i)}^T \boldsymbol{S}^T, \tag{13}$$

$$\boldsymbol{B}_i = \boldsymbol{J}_i \boldsymbol{M}^{-1} \sum_{t=1}^{i-1} \boldsymbol{N}_{pre(t)}^T \boldsymbol{S}^T \boldsymbol{\tau}_t - \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \boldsymbol{h}_{i|pre(i)}, \tag{14}$$

where $\boldsymbol{\Lambda}_{i|pre(i)}$, $\boldsymbol{\mu}_{i|pre(i)}$, $\boldsymbol{\rho}_{i|pre(i)}$ represent respectively the task i operational space inertia, Coriolis/centrifugal, gravity forces which are in null spaces of tasks $< i$, detailed in [7].

The hierarchical convex optimization can be designed for each hierarchy in the operational space simply as,

$$\boldsymbol{J}_i(\boldsymbol{\tau}_i) = \|\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i\|^2 + \varepsilon\|\boldsymbol{\tau}_i\|^2 \tag{15a}$$

$$= \tfrac{1}{2}\boldsymbol{\tau}_i^T(\boldsymbol{A}_i\boldsymbol{A}_i^T + \varepsilon\mathbf{1})\boldsymbol{\tau}_i + \boldsymbol{\tau}_i^T\boldsymbol{A}_i(\boldsymbol{B}_i - \ddot{\mathbf{x}}_i^*) \tag{15b}$$

$$= \tfrac{1}{2}\boldsymbol{\tau}_i^T\boldsymbol{H}_i\boldsymbol{\tau}_i + \boldsymbol{\tau}_i^T\boldsymbol{g}_i, \tag{15c}$$

where $\boldsymbol{J}_i(\boldsymbol{\tau}_i)$ is the cost function. $\boldsymbol{H}_i$ and $\boldsymbol{g}_i$ denote the symmetric Hessian matrix and the gradient vector for hierarchy i. The objective is regularized by a small value ε, which ensures positive definiteness of the objective Hessian. $\boldsymbol{\tau}_i$ is the extra torque required apart from the higher level ones.

We define the optimal actuated torque for hierarchies $\leq i$ as $\boldsymbol{\tau}_{opt|i}$ and we propose to use QR decomposition projector $\boldsymbol{P}_{QR}$ as a mapping matrix in the hierarchical optimization,

$$\boldsymbol{\tau}_{opt|i} = \boldsymbol{P} \sum_{t=1}^{i} \boldsymbol{N}_{pre(t)}^T \boldsymbol{S}^T \boldsymbol{\tau}_t, \tag{16}$$

where $\boldsymbol{P} = (\boldsymbol{P}_{QR}\boldsymbol{S}^T)^{+}\boldsymbol{P}_{QR}$ and $\boldsymbol{P}_{QR}$ only depends on the contact Jacobian and ensures the actuated torque to be independent on the contact force [16]. In each optimization, $\boldsymbol{\tau}_{opt|i-1}$ is the torque required for $< i$ hierarchies, which should be integrated into the inequality constraints in Section III-E. In the final hierarchy K, the required final torque for all hierarchies $\boldsymbol{\tau}_{opt|K}$ is achieved. With the proposed recursively updated dynamics model and null-space projections, the hierarchical scheme avoids equality constraints. To achieve task compliance, the model can be modified to (29) to include our prioritized impedance controllers in Section III-D.

C. New Integrated Prioritized Weighted Framework

When tasks in some hierarchies need importance distribution and no priority can be decided, the weighted optimization should be applied. Assume that there are r_i tasks in hierarchy i and they should all be in the null space of higher-hierarchy tasks.

In contrast with the conventional weighted optimization, we propose to use null-space projections among the r_i sub-tasks as if hierarchies exist among them. Although this weighted scheme cannot ensure strict hierarchies, when there is not enough DOF for the robot, feasible motion directions can be selected for one specific task in this hierarchy by using decomposition methods. In addition, to achieve compliance of the sub-tasks, this scheme enables to integrate impedance controllers and decouple its influence to each sub-task. This is the first time that the prioritized weighted scheme is integrated into the hierarchical convex optimization and the quadratic-form extractions of r_i sub-tasks are achieved.

The Jacobian for the r_i sub-tasks can be written as: $\boldsymbol{J}_i = \begin{bmatrix} \boldsymbol{J}_{i_1}^T & \cdots & \boldsymbol{J}_{i_{r_i}}^T \end{bmatrix}^T$, where $\boldsymbol{J}_{i_j}$ is the Jacobian for sub-task j in hierarchy i. Assume that the previous $(i-1)$ hierarchies have been calculated, the r_i sub-tasks in this hierarchy will be optimized together. We define $\ddot{\mathbf{x}}_i$ to combine all sub-tasks and each task acceleration $\ddot{\mathbf{x}}_{i_j}$ can be extracted depending on the actuated torque using the updated dynamics model in (11) as,

$$\ddot{\mathbf{x}}_{i_j} = \boldsymbol{A}_{i_j}^T \boldsymbol{\tau}_i + \boldsymbol{B}_{i_j}, \ j \in \{1, \cdots, r_i\} \tag{17}$$

where the coefficient matrices $\boldsymbol{A}_{i_j}$ and $\boldsymbol{B}_{i_j}$ are derived as,

$$\boldsymbol{A}_{i_j}^T = \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1}(\boldsymbol{A}_{F_{i_j}}^T + \boldsymbol{A}_{T_{i_j}}^T), \tag{18}$$

$$\boldsymbol{B}_{i_j} = \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1}(\boldsymbol{B}_{F_{i_j}} + \boldsymbol{B}_{T_{i_j}} - \boldsymbol{h}_{i_j|pre(i_j)}), \tag{19}$$

where the middle-process variables ($\boldsymbol{A}_{F_{i_j}}$, $\boldsymbol{B}_{F_{i_j}}$, $\boldsymbol{A}_{s_{i_j}}$, $\boldsymbol{B}_{s_{i_j}}$, $\boldsymbol{A}_{T_{i_j}}$ and $\boldsymbol{B}_{T_{i_j}}$) are used to calculate $\boldsymbol{A}_{i_j}$ and $\boldsymbol{B}_{i_j}$ in a recursive way,

shown in Algorithm 2. $\boldsymbol{J}_{i_j|pre(i_j)}$ means that the sub-task j is in the null spaces of the $i-1$ hierarchical tasks and $j-1$ sub-tasks in hierarchy i. Then the r_i task references can be combined as below,

$$\ddot{\mathbf{x}}_i = \begin{bmatrix} \ddot{\mathbf{x}}_{i_1} \\ \vdots \\ \ddot{\mathbf{x}}_{i_{r_i}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{i_1}^T \boldsymbol{\tau}_i + \boldsymbol{B}_{i_1} \\ \vdots \\ \boldsymbol{A}_{i_{r_i}}^T \boldsymbol{\tau}_i + \boldsymbol{B}_{i_{r_i}}. \end{bmatrix} = \boldsymbol{A}_i^T \boldsymbol{\tau}_i + \boldsymbol{B}_i. \tag{20}$$

Then the prioritized weighted convex optimization can be integrated into the hierarchical framework and the cost function $\boldsymbol{J}_i(\boldsymbol{\tau}_i)$ in (15) becomes as follows,

$$\boldsymbol{J}_i(\boldsymbol{\tau}_i) = \|\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i\|_{\boldsymbol{w}_i}^2 + \varepsilon\|\boldsymbol{\tau}_i\|^2 \tag{21a}$$

$$= \tfrac{1}{2}\boldsymbol{\tau}_i^T(\boldsymbol{A}_i\boldsymbol{A}_i^T\boldsymbol{w}_i + \varepsilon\mathbf{1})\boldsymbol{\tau}_i + \boldsymbol{\tau}_i^T\boldsymbol{A}_i\boldsymbol{w}_i(\boldsymbol{B}_i - \ddot{\mathbf{x}}_i^*) \tag{21b}$$

$$= \tfrac{1}{2}\boldsymbol{\tau}_i^T\boldsymbol{H}_i\boldsymbol{\tau}_i + \boldsymbol{\tau}_i^T\boldsymbol{g}_i, \tag{21c}$$

where $\boldsymbol{w}_i$ is the weight matrix. Then (16) is applied to allow the optimized torque to combine the hierarchies $\leq i$. When singularity exists for one specific sub-task j in this hierarchy, this prioritized weighted scheme enables us to apply the eigen-decomposition method in [7] to select the feasible motion directions by changing $\boldsymbol{\Lambda}_{i_j|pre(i_j)}$ as below,

$$\boldsymbol{\Lambda}_{i_j|pre(i_j)} = \boldsymbol{U}_{r(i_j)}\boldsymbol{\Sigma}_{r(i_j)}^{-1}\boldsymbol{U}_{r(i_j)}^T, \tag{22}$$

where $\boldsymbol{\Sigma}_{r(i_j)}$ is a diagonal matrix of non-zero eigenvalues. $\boldsymbol{U}_{r(i_j)}$ is the matrix corresponding to non-zero eigenvectors. In addition, the scheme is more advantageous, our prioritized impedance controllers in Section III-D can be integrated to influence the sub-tasks in a decoupled way. The hierarchical and prioritized weighted scheme is calculated in Algorithm 2 by using the middle-process variables, $\boldsymbol{L}_i$ and $\boldsymbol{V}_i$.

D. Proposed Prioritized Impedance Controllers

In this subsection, prioritized impedance controllers are proposed and developed to be compatible with the hybrid optimization framework. In other words, various impedance forces should be consistent with the task hierarchies.

For our robot, we control three operational hierarchical tasks (see Fig. 4), including the highest hierarchy for the quadruped-CoM motion signed as G, the second hierarchy for the wheeled motion signed as w, and the lowest hierarchy for the manipulator motion signed as m. To verify our hierarchical-weighted framework, the task G is artificially divided into two sub-tasks, including the translational motion task signed as G_v and the angular motion task signed as G_ω, and then the prioritized weighted scheme is applied.

For the two CoM-motion sub-tasks, two impedance controllers can be integrated with priorities as below,

$$\boldsymbol{\tau}_{im_{G_v}} = \boldsymbol{J}_{G_v}^T\{\boldsymbol{K}_{G_v}(\boldsymbol{p}_G^d - \boldsymbol{p}_G) + \boldsymbol{D}_{G_v}(\boldsymbol{v}_G^d - \boldsymbol{v}_G)\} \in \mathbb{R}^n, \tag{23a}$$

$$\boldsymbol{\tau}_{im_{G_\omega}} = \boldsymbol{J}_{G_\omega|G_v}^T\{\boldsymbol{D}_{G_\omega}(\boldsymbol{k}_G^d - \boldsymbol{k}_G)\} \in \mathbb{R}^n, \tag{23b}$$

where $\boldsymbol{K}_{G_v}$ and $\boldsymbol{D}_{G_v}$ are the stiffness and damping matrices respectively for the quadruped-CoM translational motion. Since the quadruped-CoM angular position has no sense, only the impedance controller on the angular momentum $\boldsymbol{k}_G$ is considered and $\boldsymbol{D}_{G_w}$ is the damping matrix. $\boldsymbol{J}_{G_v}$ and $\boldsymbol{J}_{G_\omega}$

are the Jacobians related to the translational and angular quadruped-CoM motions, respectively. $G_\omega|G_v$ means that the sub-task (G_ω) is in the null space of sub-task (G_v).

For the wheel locomotion, the rolling resistance is not modeled and one impedance controller is built to compensate the inaccuracy of the contact model,

$$\boldsymbol{\tau}_{im_w} = \boldsymbol{S}_w^T\{\boldsymbol{K}_w(\boldsymbol{q}_w^d - \boldsymbol{q}_w) + \boldsymbol{D}_w(\dot{\boldsymbol{q}}_w^d - \dot{\boldsymbol{q}}_w)\} \in \mathbb{R}^n, \quad (24)$$

where $\boldsymbol{S}_w$ is the selection matrix mapping vectors from $\mathbb{R}^{n_w}$ to $\mathbb{R}^n$ dimension. In this subsection, $\boldsymbol{K}_t$ and $\boldsymbol{D}_t$ always denote the stiffness and damping matrices respectively with different subscripts, for this case, t represents the wheel motion w.

The manipulator enables the robot to achieve various missions. One impedance controller is added at its end effector to achieve manipulation compliance as follows,

$$\boldsymbol{\tau}_{im_m} = \boldsymbol{J}_m^T\{\boldsymbol{K}_m(\boldsymbol{q}_m^d - \boldsymbol{q}_m) + \boldsymbol{D}_m(\dot{\boldsymbol{q}}_m^d - \dot{\boldsymbol{q}}_m)\} \in \mathbb{R}^n, \quad (25)$$

where $\boldsymbol{q}_m$ is the 6-DOF manipulator joint position and is achieved by inverse kinematics. $\boldsymbol{J}_m$ is its relative Jacobian.

In addition, one impedance controller is designed to allow the base with small angular motion as below,

$$\boldsymbol{\tau}_{im_b} = \boldsymbol{J}_b^T\{\boldsymbol{K}_b(\boldsymbol{\theta}_b^d - \boldsymbol{\theta}_b) + \boldsymbol{D}_b(\boldsymbol{\omega}_b^d - \boldsymbol{\omega}_b)\} \in \mathbb{R}^n, \quad (26)$$

where $\boldsymbol{J}_b$ is the Jacobian for the base angular motion.

Furthermore, another impedance controller is established to active legged suspension as follows,

$$\boldsymbol{\tau}_{im_g} = \boldsymbol{J}_g^T\{\boldsymbol{K}_g(\boldsymbol{p}_g^d - \boldsymbol{p}_g) + \boldsymbol{D}_g(\dot{\boldsymbol{p}}_g^d - \dot{\boldsymbol{p}}_g)\} \in \mathbb{R}^n, \quad (27)$$

where $\boldsymbol{p}_g \in \mathbb{R}^4$ combines all wheel-center positions in the base frame and its desired value is generated from the legged motion generator. $\boldsymbol{J}_g$ is the relative Jacobian.

We propose to integrate prioritized impedance controllers into this hybrid control framework. All of them should be consistent with task priorities and we define the prioritized sum of the impedance control torque as below,

$$\boldsymbol{\Gamma}_i = \{\sum_{t=1}^{i} \boldsymbol{J}_{t|pre(t)}^T \bar{\boldsymbol{J}}_{t|pre(t)}^T\} \sum \boldsymbol{\tau}_{im_k} \in \mathbb{R}^n, \quad (28)$$

where $\sum \boldsymbol{\tau}_{im_k}$ denotes the sum of the impedance controllers, shown in Fig. 4, including the ones relating to the specific designed tasks in (23), (24) and (25), and the ones not related to the designed tasks in (26) and (27) and other impedance controllers, e.g. self-collision avoidance. Then we integrate $\boldsymbol{\Gamma}_i$ into the recursively updated dynamics model as follows,

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C} + \boldsymbol{G} = \boldsymbol{J}_c^T\boldsymbol{F}_c + \sum_{t=1}^{i-1} \boldsymbol{N}_{pre(t)}^T\boldsymbol{S}^T\boldsymbol{\tau}_t + \boldsymbol{S}^T\boldsymbol{\tau}_i - \boldsymbol{\Gamma}_i, \quad (29)$$

By using this model, $\boldsymbol{B}_i$ and $\boldsymbol{B}_{i_j}$ in (14) and (19) should be modified correspondingly, shown in Algorithm 2. In this way, compared to the conventional hierarchical and weighted schemes which can integrate impedance controllers relating to the designed tasks, our approach has two more advantages:

- Prioritized impedance forces influence the designed tasks in a decoupled way, especially for sub-tasks in the weighted scheme.
- General impedance controllers indirectly acting on the designed tasks can also be integrated with priorities.

Algorithm 2: Compact Form Dynamics Controller

Initialization: $\boldsymbol{\tau}_{opt|0} = \boldsymbol{0}$, $\boldsymbol{\Gamma}_0 = \boldsymbol{0}$.
Middle-process variables: $\boldsymbol{L}_j$, $\boldsymbol{V}_j$, $j \in \{0, \cdots, K\}$, and $\boldsymbol{L}_0 = \boldsymbol{V}_0 = \boldsymbol{0}$.
for $i=1$ to K **do**
  if hierarchy i has $r_i = 1$ task **then**
    $\boldsymbol{A}_i^T = \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \bar{\boldsymbol{J}}_{i|pre(i)}^T \boldsymbol{S}^T$
    $\boldsymbol{B}_i = \boldsymbol{J}_i \boldsymbol{M}^{-1} \boldsymbol{L}_{i-1} - \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \boldsymbol{h}_{i|pre(i)}$
    $\boldsymbol{B}_i = \boldsymbol{B}_i - \boldsymbol{J}_{i|pre(i)} \boldsymbol{M}^{-1} \boldsymbol{\Gamma}_i - \boldsymbol{J}_i \boldsymbol{M}^{-1} \boldsymbol{\Gamma}_{i-1}$
    $\boldsymbol{\tau}_i = \arg\min_{\boldsymbol{\tau}_i} \boldsymbol{J}_i(\boldsymbol{\tau}_i)$ in (15), s.t. (34)
  else
    Initialization: $\boldsymbol{A}_{s_{i_1}} = \boldsymbol{0}$, $\boldsymbol{B}_{s_{i_1}} = \boldsymbol{0}$, $\boldsymbol{A}_{T_{i_1}} = \boldsymbol{0}$, $\boldsymbol{B}_{T_{i_1}} = \boldsymbol{0}$.
    for $j = 1$ to r_i sub-task **do**
      $\boldsymbol{A}_{F_{i_j}}^T = \bar{\boldsymbol{J}}_{i_j|pre(i_j)}^T \boldsymbol{S}^T$ and $\boldsymbol{B}_{F_{i_j}} = \bar{\boldsymbol{J}}_{i_j|pre(i_j)}^T \boldsymbol{L}_{i-1}$
      if $j \neq 1$ **then**
        $\boldsymbol{A}_{s_{i_j}}^T = \boldsymbol{A}_{s_{p(i_j)}}^T + \boldsymbol{J}_{p(i_j)|pre(p(i_j))}^T \boldsymbol{A}_{F_{p(i_j)}}^T$
        $\boldsymbol{B}_{s_{i_j}} = \boldsymbol{B}_{s_{p(i_j)}} + \boldsymbol{J}_{p(i_j)|pre(p(i_j))}^T \boldsymbol{B}_{F_{p(i_j)}}$
        $\boldsymbol{A}_{T_{i_j}}^T = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \boldsymbol{J}_{i_j} \boldsymbol{M}^{-1} \boldsymbol{A}_{s_{i_j}}^T$
        $\boldsymbol{B}_{T_{i_j}} = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \boldsymbol{J}_{i_j} \boldsymbol{M}^{-1}(\boldsymbol{B}_{s_{i_j}} + \boldsymbol{V}_{i-1})$
      end
      $\boldsymbol{A}_{i_j}^T = \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1}(\boldsymbol{A}_{F_{i_j}}^T + \boldsymbol{A}_{T_{i_j}}^T)$
      $\boldsymbol{B}_{i_j} = \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1}[\boldsymbol{B}_{F_{i_j}} + \boldsymbol{B}_{T_{i_j}} - \boldsymbol{h}_{i_j|pre(i_j)}] - \boldsymbol{J}_{i_j|pre(i_j)} \boldsymbol{\Gamma}_i$
      $\boldsymbol{B}_{i_j} = \boldsymbol{B}_{i_j} - \boldsymbol{J}_{i_j} \boldsymbol{M}^{-1}\{\boldsymbol{\Gamma}_{i-1} + \sum_{t=1}^{j-1} \boldsymbol{J}_{i_t|pre(i_t)}^T \bar{\boldsymbol{J}}_{i_t|pre(i_t)}^T \boldsymbol{\Gamma}_i\}$
    end
    $\boldsymbol{A}_i = [\boldsymbol{A}_{i_j}, \cdots, \boldsymbol{A}_{i_{r_i}}]$ and $\boldsymbol{B}_i = [\boldsymbol{B}_{i_1}^T, \cdots, \boldsymbol{B}_{i_{r_i}}^T]^T$
    $\boldsymbol{\tau}_i = \arg\min_{\boldsymbol{\tau}_i} \boldsymbol{J}_i(\boldsymbol{\tau}_i)$ in (21), s.t. (34)
  end
  $\boldsymbol{L}_i = \boldsymbol{L}_{i-1} + \boldsymbol{N}_{pre(i)}^T \boldsymbol{S}^T \boldsymbol{\tau}_i$ and $\boldsymbol{V}_i = \boldsymbol{V}_{i-1} + \boldsymbol{J}_{i|pre(i)}^T \bar{\boldsymbol{J}}_{i|pre(i)}^T \boldsymbol{L}_i$
  $\boldsymbol{\tau}_{opt|i} = \boldsymbol{P}\boldsymbol{L}_i$
end

E. Recursive Updated Inequality Constraints

The joint limits, actuation limits, unilateral condition and the friction limitations are as follows,

$$\boldsymbol{q}_{g|min} \leq \boldsymbol{q}_{g|k+1} = \boldsymbol{q}_{g|k} + \dot{\boldsymbol{q}}_{g|k}\delta t + \tfrac{1}{2}\ddot{\boldsymbol{q}}_{g|k}\delta t^2 \leq \boldsymbol{q}_{g|max}$$
$$\dot{\boldsymbol{q}}_{g|min} \leq \dot{\boldsymbol{q}}_{g|k+1} = \dot{\boldsymbol{q}}_{g|k} + \ddot{\boldsymbol{q}}_{g|k}\delta t \leq \dot{\boldsymbol{q}}_{g|max}$$
$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_{|k} \leq \boldsymbol{\tau}_{max} \quad (30)$$
$$\boldsymbol{u}_c \cdot \boldsymbol{F}_c \geq \boldsymbol{0}$$
$$|\boldsymbol{t}_c \cdot \boldsymbol{F}_c| \leq \tfrac{\mu}{\sqrt{2}}(\boldsymbol{u}_c \cdot \boldsymbol{F}_c) \text{ and } |\boldsymbol{b}_c \cdot \boldsymbol{F}_c| \leq \tfrac{\mu}{\sqrt{2}}(\boldsymbol{u}_c \cdot \boldsymbol{F}_c),$$

where $|k$ and δt denote the kth instance and the fixed sampling time, respectively. μ is the friction coefficient. $(\boldsymbol{t}_c, \boldsymbol{b}_c, \boldsymbol{u}_c)$ are used to select the normal, tangential and lateral directions of the contact force. The contact force $\boldsymbol{F}_c$ can be given by the actuated torque $\boldsymbol{\tau}$ as below,

$$\boldsymbol{F}_c = -\bar{\boldsymbol{J}}_c^T\boldsymbol{S}^T\boldsymbol{\tau} + (\boldsymbol{\mu}_c + \boldsymbol{\rho}_c) = \boldsymbol{A}_c^T\boldsymbol{\tau} + \boldsymbol{B}_c, \quad (31)$$

where $\bar{\boldsymbol{J}}_c$ is the dynamically generalized inverse of contact Jacobian. $\boldsymbol{\mu}_c$ and $\boldsymbol{\rho}_c$ are the operational space coriolis and gravity forces. The quadratic form of joint acceleration is extracted using our updated dynamics model with prioritized impedance controllers in (29) as below,

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{J}_c^T\boldsymbol{A}_c^T + \boldsymbol{S}^T)\boldsymbol{\tau} + \boldsymbol{M}^{-1}(\boldsymbol{J}_c^T\boldsymbol{B}_c - \boldsymbol{C} - \boldsymbol{G} - \boldsymbol{\Gamma}_i), \quad (32)$$

where $\boldsymbol{\Gamma}_i$ is integrated into this extraction. The constraints in (30) can be written according to the actuated torque using the extractions in (32) and (31) as below,

$$\boldsymbol{U}_{min} \leq \boldsymbol{Q}\boldsymbol{\tau} \leq \boldsymbol{U}_{max}, \quad (33)$$
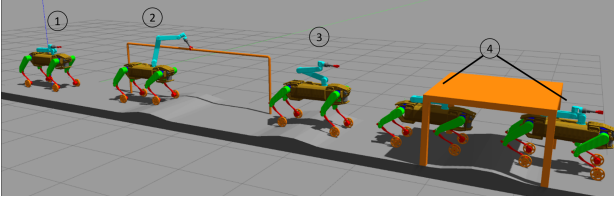
Fig. 5. The robot runs on the rough terrain and handles activities which are divided into four stages. Stage 1 shows the beginning state. In stage 2, the robot catches the handrail using the manipulator. After the handrail, the robot brings back the manipulator in stage 3. Then the robot squirts down and crosses over the last bump and through the desk in stage 4.

where Q is the coefficient matrix combining all constraints in (30). To enable (33) to be compatible with our control framework, it should be modified in a recursive way as,

$$U_{min} \leq Q(PN_{pre(i)}^T S^T \tau_i + \tau_{opt|i-1}) \leq U_{max}. \quad (34)$$

IV. NUMERICAL SIMULATION RESULTS

The control framework proposed in Section III is tested on our virtual robot, "Towr*ISIR*" in Fig. 1. The optimization is implemented using the QPOASES library, a real-time QP solver [18]. Simulation results are reported using one physical engine established by Ros-Gazebo. The null-space based theory ensures the multi-task control hierarchy, we use simulations to verify the usefulness and validity of our hybrid control framework with prioritized impedance controllers.

The first simulation (see Fig. 5) relates to the robot rolling on unknown rough terrain along with holding one handrail using the manipulator. Then the robot retracts its manipulator and squirts down to cross over the final bump and through one desk. The handrail and desk positions are known, but the terrain geometry is unknown. The quadruped-CoM motion reference is set to be in the sagittal plane.

The simulation results are shown in Fig. 6. Along the whole process, the quadruped-CoM tracking errors are very small, and the desired and actual trajectories are almost coincident as can be noticed in Fig. 6(a) and 6(b). Even when the robot squirts down to cross through one desk with an unknown bump under it, the prioritized impedance controller still enables the tracking of the CoM height reference with a small error. Since the quadruped-CoM translational motion is one sub-task in the prioritized weighted scheme (see Fig. 4), its tracking performance can also be improved by adjusting the weight matrix. The legged motion satisfies the task hierarchy and the contact constraint, which follows the CoM motion reference to achieve squirting motion. The impedance controller related to the legged motion indirectly influences the whole-body motion, it is integrated into our prioritized impedance controller which ensures the consistency with task hierarchies and enables the robot to recover to its normal state after climbing several bumps. Since we assume that the contact points are always at the lowest points of the wheels, it is not precise when the robot runs on bumps or slopes. Therefore, the wheel tracking (see Fig. 6(c)) experiences moderately bigger errors when running on bumps. In addition, it is difficult to estimate the rolling resistance force because of the unknown rolling friction. This issue also influences the outputs of the motion generator and the results of the quadruped-CoM tracking when crossing bumps in Fig. 6(a) and Fig. 6(b). However, with the prioritized impedance controllers related to the wheel motion, the final wheel motion error remains at about $0.15rad$ which is very small. For the manipulator motion, to catch the handrail all the time, its trajectory is defined in a bigger range. The position and orientation errors of the manipulator experiences a relatively big fluctuation, shown in Fig. 6(d). However, even the touching-handrail force is unknown, by using our hybrid dynamics controller integrated with our prioritized impedance forces, the robot successfully holds the handrail. In this simulation, even with the interaction uncertainties, including unknown rough terrain with bumps, imprecise contact points, unknown rolling resistance and unknown external force from the handrail,
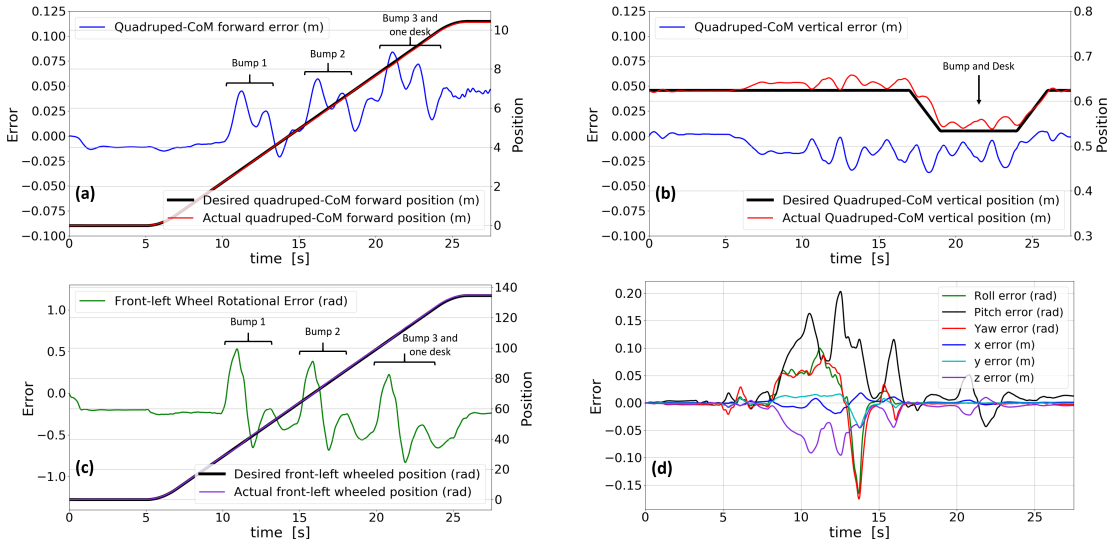

Fig. 6. Monitor of robot states. The left axes represent errors and the right axes denote positions. **(a)** and **(b)** monitor the quadruped-CoM motion in forward and vertical directions respectively. In **(c)**, we only plot the front-left wheel position and error because the four wheels show the similar results. **(d)** monitors the arm end effector motion errors, including its orientations (roll, pitch, yaw) and positions (x, y, z).
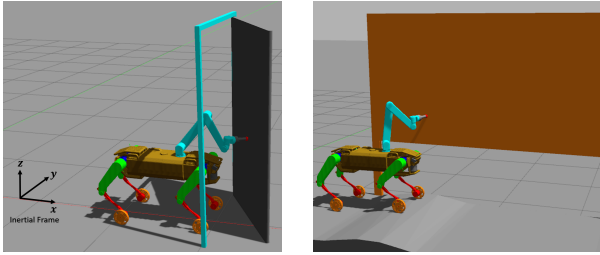
Fig. 7. Two other scenarios. [Left]: The robot is driven on flat ground and uses its manipulator to open a door. [Right]: The robot runs on rough terrain along with washing a wall.

by applying our dynamics controller, the robot successfully finishes these missions with compliance, especially tracking the quadruped-CoM motion.

Another two scenarios imitate the open-door process inside offices in Fig. 7 [Left], and washing a wall in out-door environments in Fig. 7 [Right]. The quadruped centroidal motion tracking on the flat ground is better than on the terrain with bumps. Even though the wheel rolling resistance force, the open-door force and wash-wall forces are unknown, the robot succeeds to handle the interaction uncertainties compliantly using our dynamics controller with prioritized impedance forces. The main results are shown in the attachment video.

All the simulations are done on a usual PC with a i7-7700 CPU, $3.6GHz$. The sampling time is $2ms$ and at each time step, our algorithm is solved with 22 decision variables and 86 inequality constraints. The calculation time is max $1.3ms$, about $769Hz$. The calculation speed is similar to other works, as well as the tracking performance with suitable parameters in the feedback controllers. However, by using our recursively updated dynamics model and the prioritized impedance controller, our hybrid hierarchical and prioritized weighted control scheme is more general and complete which ensures the priority consistency of various impedance forces with the tasks in two kinds of control schemes.

To summarize, the results provided by our controller are quite satisfactory. This controller enables the robot to track the quadruped-CoM motion adaptively and effectively, and handle complex duties with compliance respecting contact constraints and actuator limits.

V. CONCLUSION

In this paper, we developed a dynamics torque controller to track the whole-body behaviors of a high-DOF quadruped-on-wheel robot. The wheel motion generator enables the robot to track its centroidal motion, and the generated wheel motion is consistent with the robot legged suspension. For the torque controller, we propose the concept of the recursively updated dynamics model. It enables to integrate the proposed prioritized weighted scheme into the hierarchical framework, and enables to combine our proposed prioritized impedance controllers to ensure the consistency of the impedance forces with the task hierarchies. General impedance controllers indirectly acting on the designed tasks can also be integrated with priorities, and they influence the tasks in a decoupled way,

especially for sub-tasks in the prioritized weighted scheme. By using this model, the task accelerations in two kinds of control schemes are extracted depending on the actuated torque and the prioritized impedance forces. Our dynamics controller is more general and complete, and its algorithm is tested in three simulation scenarios. The results show that our hybrid hierarchical and prioritized weighted control framework is useful and valid, and enables the robot to track the whole-body motion references and handle various interaction uncertainties efficiently and compliantly.

REFERENCES

[1] Grand Christophe, Benamar Faïz, Plumet Frédéric, et al. Stability and traction optimization of a reconfigurable wheel-legged robot[J]. The International Journal of Robotics Research, 2004, 23(10-11): 1041-1058.

[2] Arthur Bouton, Christophe Grand and Faïz Benamar. Motion control of a compliant wheel-leg robot for rough terrain crossing. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, pages 2846-2851. IEEE, 2016.

[3] Thomson T, Sharf I, Beckman B. Kinematic control and posture optimization of a redundantly actuated quadruped robot. Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012: 1895-1900.

[4] Marko Bjelonic, C. Dario Bellicoso, Yvain de Viragh, Dhionis Sako, F. Dante Tresoldi, Fabian Jenelten and Marco Hutter, Keep Rollin' – Whole-Body Motion Control and Planning for Wheeled Quadrupedal Robots, ArXiv, 2018

[5] Wensing P M, Orin D E. Improved computation of the humanoid centroidal dynamics and application for whole-body control[J]. International Journal of Humanoid Robotics, 2016, 13(01): 1550039.

[6] Malgorzata Kamedula, Navvab Kashiri, Nikos G. Tsagarakis, On the Kinematics of Wheeled Motion Control of a Hybrid Wheeled-Legged CENTAURO robot. IROS 2018: 2426-2433

[7] Sentis L, Khatib O. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives[J]. International Journal of Humanoid Robotics, 2005, 2(04): 505-518.

[8] Bouyarmane K, Kheddar A. Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations[C]. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE, 2011: 4414-4419.

[9] Escande A, Mansard N, Wieber P B. Hierarchical quadratic programming: Fast online humanoid-robot motion generation[J]. The International Journal of Robotics Research, 2014, 33(7): 1006-1028.

[10] Ponce O E R. Generation of the whole-body motion for humanoid robots with the complete dynamics[D]. Universite Toulouse III Paul Sabatier, 2014.

[11] Wensing P M, Orin D E. Generation of dynamic humanoid behaviors through task-space control with conic optimization[C]. Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013: 3103-3109.

[12] Hutter M, Sommer H, Gehring C, et al. Quadrupedal locomotion using hierarchical operational space control[J]. The International Journal of Robotics Research, 2014, 33(8): 1047-1062.

[13] Herzog A, Rotella N, Mason S, et al. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid[J]. Autonomous Robots, 2016, 40(3): 473-491.

[14] De Lasa M, Mordatch I, Hertzmann A. Feature-based locomotion controllers[C]. ACM Transactions on Graphics (TOG). ACM, 2010, 29(4): 131.

[15] Featherstone, Roy. Rigid body dynamics algorithms. Springer, 2014.

[16] Righetti L, Buchli J, Mistry M, et al. Inverse dynamics control of floating-base robots with external constraints: A unified view[C]. Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011: 1085-1090.

[17] Walker M W, Orin D E. Efficient dynamic computer simulation of robotic mechanisms[J]. Journal of Dynamic Systems, Measurement, and Control, 1982, 104(3): 205-211.

[18] Ferreau H J, Kirches C, Potschka A, et al. qpOASES: A parametric active-set algorithm for quadratic programming[J]. Mathematical Programming Computation, 2014, 6(4): 327-363.

[19] Smits, R. KDL: Kinematics and Dynamics Library, http://www.orocos.org/kdl