



**HAL**  
open science

# Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics

Wenqian Du, Mohamed Fnadi, Faïz Ben Amar

## ► To cite this version:

Wenqian Du, Mohamed Fnadi, Faïz Ben Amar. Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics. *Mechanism and Machine Theory*, 2020, 153, pp.103984. 10.1016/j.mechmachtheory.2020.103984 . hal-03177960

**HAL Id: hal-03177960**

**<https://hal.science/hal-03177960v1>**

Submitted on 24 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Rolling based Locomotion on Rough Terrain for a Wheeled Quadruped using Centroidal Dynamics

Wenqian Du\*, Mohamed Fnadi and Faiz Benamar

Sorbonne University, CNRS, Institut des Systèmes Intelligents et de Robotique: ISIR, F75005, Paris, France

---

## ARTICLE INFO

### Keywords:

motion generation  
quadruped robot  
wheeled robot  
centroidal dynamics  
kinematics

## ABSTRACT

This paper proposes a new wheel motion generator to track the centroidal motion of one quadruped-on-wheel robot which has the ability to cross various rough terrains with the model based whole-body torque control. The generator is used to track the whole-robot centroidal motion reference. Firstly, the wheel contact model and the whole-body inverse kinematics model are derived using spatial vectors. The wheel motion is extracted out mathematically depending on the base and the legged motions, which serves as the kinematics model. Then the wheel motion generator is developed by combining both the kinematics model and the robot centroidal momentum/dynamics model. The models are decomposed into three components relating to the base motion, the legged motion and the wheel motion. The adaptive wheel motion references are derived in a detailed mathematical way and several algorithms are developed for the model decompositions. Finally, the robot is simulated to be driven on various rough terrains using the operational space control framework mixed with our proposed compatible prioritized impedance controller. The required torque for multiple tasks is generated by the feed-forward and feedback controllers while fulfilling the contact constraints.


---

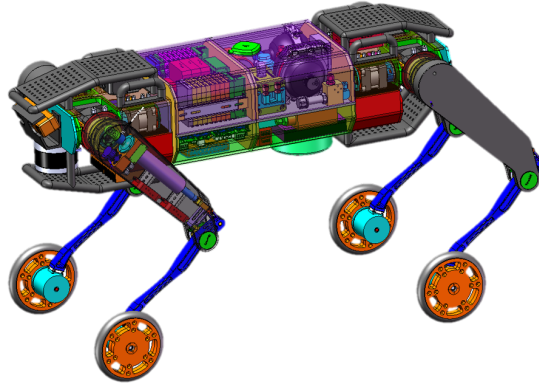
## 1. Introduction

Conventional wheeled robots have great performance and potential on continuous terrains with high speeds and less energy consuming, but easily stuck on the rough terrain with steep obstacles. In contrast, legged robots are not tied to the discontinuous environments like mountain lanes, or artificial stairs, and can move with agility to cope with uncertainties or disturbances. From a practical point of view, quadruped-on-wheel robots combine both strengths and have more versatile locomotion modes from rolling to walking, climbing, running and jumping, etc. However, these enhanced locomotion skills come with a more complex dynamics model, motion generation and torque control.

Here, only quadruped-on-wheel robots with active redundant actuators are reviewed here. The works in the state of the art always track the robot CoM motion which is one special point to represent the whole-robot motion. The literature presents various methods to evaluate the robot balance that includes the center of mass (CoM) position, the zero moment point (ZMP) and the center of pressure [1]. Recently, the centroidal angular velocity or momentum is also tracked in addition to the CoM translational motion for highly redundant robots, e.g. humanoids and quadrupeds. The system translational and angular momentum can be defined here naturally [2] and the 6-dimensional centroidal momentum is their combination and it is refocused as the sum of the individual link momentum expressed at the system CoM using spatial notations [2]. Authors in [3] showed that the control of the centroidal angular momentum leads to more natural rich whole-body motion of robots. Authors in [4][5] apply the control of the centroidal angular momentum to keep the balance of humanoids. Ugurlu *et al.* in [6] achieved the quadruped trot-walking by using the virtual admittance control of the whole-body angular momentum to keep postural balance. The Hylos robot [7] is composed of four wheel-legs and is capable to run on rough terrain by changing its robot CoM position and optimizing the joint torques and contact forces due to the structure redundancy, which can maximize the robot stability, however, the lack of a whole-body dynamics model hinders the robot to perform dynamic locomotion. The Complios robot is designed with active and passive compliant joints, and they are driven by series-elastic actuators (SEA) which can react to obstacles, however, the SEAs' low control bandwidth and the passive joints limit the robot speed on rough terrain. Authors in [9] developed a hydraulic actuated legged-on-wheel robot which has the similar control method with Hylos that computes the inverse kinematics to track the desired base trajectory and optimizes the posture redundantly. The Momaro robot in [10] shows one hybrid locomotion mode to overcome obstacles like stairs using a

---

 wenqian.du@isir.upmc.fr (W. Du); fnadi@isir.upmc.fr (M. Fnadi); amar@isir.upmc.fr (F. Benamar)  
ORCID(s): 0000-0002-3352-0809 (W. Du); 0000-0001-9593-8859 (M. Fnadi); 0000-0002-4590-3452 (F. Benamar)



**Figure 1:** “Towr/SIR”–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 the hip abduction/adduction joint (HAA), the hip flexion/extension joint (HFE), the knee flexion/extension joint (KFE) and the wheel forward/backward joint (WFB). The joints with hydraulic actuators apply the similar structure with the robot HyQ2Max [34] which has different leg layout and no wheels. Each joint is integrated with the force/torque sensor and the absolute/incremental encoder.

kinematics approach. However, in above works [7, 8, 9, 10], the systems show only static maneuvers, and the simplified dynamics model or the kinematics-based controller limits the diversity of dynamic behaviors. Authors in [11][12] used the ZMP index to generate the robot centroidal translational motion pattern to follow the ZMP reference in 2D and 3D space respectively. [13] extended the work in [12] and achieve the contact kinematics in the acceleration level. However, the legs in above works are always assumed to be massless, this can not be accepted since the robot CoM should combine all body mass and inertias, which constitute the whole-body dynamics model and influence the motion generation, and the wheel motion can conflict the legged suspension. Recently, one legged-wheeled centaur-like robot, CENTAURO [14][15] is developed to track the system centroidal motion and the legged-wheeled motion kinematics is achieved without the leg-massless assumption, however, this work only dealt with the joint velocity and position references which only enable to use the feedback torque controller without applying the whole-body dynamic models. In addition, the derivative of the kinematics contact constraints cannot be ensured without one feed-forward torque controller [16]. Authors in [17] proposed one multi-task operational space control framework for humanoid robots to enable the required torque for all tasks to be consistent with the contact constraints in the velocity and acceleration levels. [18] addressed the unified control of multi-contact constraints, centroidal motion maneuvers, operational tasks, postures, and internal forces for a humanoid. Instead of using the ZMP criteria with contact-coplanar limits and other disadvantages stated in [21], authors in [19][20][21] show that the satisfaction of the contact constraints in the kinematics first and second orders enable quadruped and humanoid robots with dynamic stability and feasible motions. The contact constraints can be ensured by using one optimization based torque controller which can handle the contact unilateral limit and the friction cone, and this method can also be applied to quadruped-on-wheel robots. In these torque controllers, the feedback force/torque for each designed task is integrated and consistent with the contact constraints. However, for the general feedback controllers for safety and physical constraints, such as self-collision and obstacle-avoidance missions [22], they should also be paid attention to respect the contact constraints. In this paper, we track the spatial centroidal motion for our legged-wheeled robot without leg-massless assumption, and its angular motion reference is generated in the running process. The kinematics and dynamics models are established without any simplification. Then we achieve the kinematics contact constraints in the acceleration level by using spatial vectors. Then we step further and derive the relationship in the acceleration level between the centroidal motion and all wheel-leg motions, and the centroidal momentum and dynamics models should be built for our quadruped-on-wheel robot. Then the wheel velocity and acceleration are generated on-line and they are ensured to be compatible with the actual legged suspension. In addition, we use whole-body torque controller (including the feed-forward and various feedback impedance controllers) to achieve operational-space tasks by respecting the contact constraints. The whole control framework is more general and it avoids the relative limits from each reference. This paper serve as the basis of our processing works, including the altitude control and the trotting locomotion of our quadruped-on-wheel robot, in which the control of the relative motion between the centroidal and base spatial motions enables the robot to evolve

**Table 1**

Symbols for the whole-body kinematics model, the wheel motion generators, the torque controller in Section 2, 3, 4.

${}^c_i \mathbf{X}_{w_i}$	Transformation matrix for spatial vectors from the wheel $i$ frame to the contact point $i$ frame.
${}^{w_i} \mathbf{X}_{k_i}$	Transformation matrix for spatial vectors from the knee $i$ frame to the wheel $i$ frame.
${}^c_i \mathbf{X}_{k_i}$	Transformation matrix for spatial vectors from the knee $i$ frame to the contact point $i$ frame.
${}^{w_i} \mathbf{X}_b$	Transformation matrix for spatial vectors from the base frame to the wheel-joint $i$ frame.
${}^b \mathbf{X}_G$ or ${}^1 \mathbf{X}_G$	Transformation matrix for spatial vectors from the robot CoM frame to the base frame.
${}^j \mathbf{X}_G$	Transformation matrix for spatial vectors from the robot CoM frame to the body $j$ frame.
${}^j \mathbf{X}_b$ or ${}^j \mathbf{X}_1$	Transformation matrix for spatial vectors from the base frame to the body $j$ frame.
${}^j \mathbf{X}_{p(j)}$	Transformation matrix for spatial vectors from the body $j$ parent frame to the body $j$ frame.
$\mathbf{J}_{c_i}$	Jacobian at the wheel contact $i$ point.
$\mathbf{J}_c$	Jacobian relating to all wheel-contact point velocity.
$\mathbf{J}_{k_i}$	Jacobian relating to the knee joint velocity.
$\mathbf{J}_G$	Jacobian relating to the centroidal motion.
$\mathbf{v}_{c_i}, \mathbf{v}_{c_i}, \boldsymbol{\omega}_{c_i}$	Contact-point $i$ 6-dimensional spatial velocity, 3-dimensional translational and angular velocities.
$\mathbf{v}_c$	All contact-point translational velocity.
$\mathbf{v}_b, \mathbf{v}_b, \boldsymbol{\omega}_b$	Floating-base 6-dimensional spatial velocity, 3-dimensional translational and angular velocities.
$\boldsymbol{\theta}_b$	Floating-base orientation angle.
$\mathbf{v}_G, \mathbf{v}_G, \boldsymbol{\omega}_G$	Centroidal 6-dimensional spatial velocity, 3-dimensional translational and angular velocities.
$\mathbf{v}_{w_i}$	Wheel-body $i$ spatial velocity.
$\mathbf{v}_{w_i[z]}$	Third component of $\mathbf{v}_{w_i}$ , namely the wheel angular velocity around the $z_{w_i}$ axis.
$\mathbf{v}_{k_i}$	Knee-body $i$ spatial velocity.
$\mathbf{v}_G, \boldsymbol{\omega}_G$	Centroidal translational and angular velocities respectively.
$\mathbf{q}_{w_i}, \mathbf{q}_w$	Wheel $i$ joint position, and all wheel joint position.
$\mathbf{q}_{r_i}, \mathbf{q}_r$	Leg $i$ revolute joint positions, and all leg-revolute joint position, except the wheel joints.
$\Phi_{c_i}, \Phi_{w_i}$	Free mode matrices of the virtual contact joint and the leg $i$ wheel joint respectively.
$\Phi_1, \Phi_j$	Free modes matrices of the floating-base joint and the actuated-joint $j$ .
$\mathbf{I}_G, \mathbf{I}_b^C$	Composite rigid-body inertias at the CoM and at the base respectively.
$\mathbf{I}_{G\omega}, \mathbf{I}_{Gv}$	Left and right three columns of $\mathbf{I}_G$ .
$\mathbf{I}_j, \mathbf{v}_j$	Body $j$ spatial inertia and spatial velocity respectively.
$\mathbf{I}_{w_i}$	Wheel $i$ spatial inertia.
$\mathbf{I}_b$ or $\mathbf{I}_1$	Base spatial inertia.
$\mathbf{I}_{b\omega}, \mathbf{I}_{bv}$	Left and right three columns of $\mathbf{I}_b$ .
$\mathbf{h}_G, \mathbf{p}_G, \mathbf{k}_G, \mathbf{l}_G$	Centroidal momentum, position, centroidal translational and angular momentums respectively.
$\mathbf{J}_{G c}$	Centroidal motion Jacobian in the null-space of the contact constraint.
$\mathbf{J}_{w G c}$	Wheel motion Jacobian in the null-space of the centroidal motion and the contact constraint.
$\mathbf{N}_c$	Null-space projector for the contact constraint.
$\mathbf{N}_{G c}$	Null-space projector combining the centroidal motion task and the contact constraint.
$\mathbf{N}_{all}$	Null-space projector combining all tasks and the contact constraint.
$\bar{\mathbf{J}}$	Dynamically inverse of any Jacobian.
$\mathbf{F}_c, \boldsymbol{\mu}_c, \boldsymbol{\rho}_c, \boldsymbol{\Lambda}_c$	Operational force, Coriolis and centripetal force, gravity force and inertia at the contact points.
$\mathbf{F}_{G c}$	Prioritized operational space force for the centroidal motion task.
$\boldsymbol{\mu}_{G c}, \boldsymbol{\rho}_{G c}, \boldsymbol{\Lambda}_{G c}$	Prioritized Coriolis and centripetal force, gravity force and inertia for the centroidal motion task.
$\mathbf{F}_{w G c}$	Prioritized operational space force for the wheel motion task.
$\boldsymbol{\mu}_{w G c}, \boldsymbol{\rho}_{w G c}, \boldsymbol{\Lambda}_{w G c}$	Prioritized Coriolis and centripetal force, gravity force and inertia for the wheel motion task.
$\sum \boldsymbol{\tau}_k$	Sum of all impedance torques.
$\boldsymbol{\tau}_{fb}, \boldsymbol{\tau}, \boldsymbol{\tau}_{act}$	Feedback torque and feed-forward torque, and their combination.

on rough terrains without pre-defined legged trajectories.

The derivative of the centroidal momentum has been introduced as the centroidal dynamics [23] which is extended by a coordinate transformation that produces an inertially decoupled structure of the equations of motion for a floating base robot [24]. Furthermore, the centroidal momentum matrix and the centroidal momentum bias force are extracted out from the generalized dynamics model [3] which improves the calculation of the centroidal dynamics properties. Authors in [25] modifies the closed-loop momentum-based controller structure to avoid instability of the zero dynamics

which enable the asymptotic stability. The centroidal dynamics model is applied in [26] to compute a feasible CoM trajectory through the mapping of contact forces. In [27], the capture point concept is implemented and its references are generated by one momentum-based torque controller which provides one humanoid robot with abilities of push recovery and reaction to other external disturbances.

In this paper, we focus on the rolling based locomotion and propose a new adaptive wheel motion generator to track the whole-body centroidal motion reference of one quadruped-on-wheel robot, and present a set of scientific theoretical contributions for this purpose. This new wheel motion generator combines the whole-body inverse kinematics model and the robot centroidal momentum/dynamics model without leg-massless assumption. All wheels are maintained in contact with the ground, and the legged motion is used for the legged active suspension and for the base posture control. The novel wheel motion generator is achieved by respecting the non-slippage contact constraints and the actual legged suspension in the velocity and acceleration levels, under the goal to track the centroidal motion reference. The wheel contact constraints are derived by using spatial vectors and the wheel motion is extracted out mathematically depending on the base and the legged motions, which serves as the kinematics model. Then the kinematics and centroidal dynamics models are decomposed into several components relating to the base, the leg and the wheel motions. Several algorithms are developed for the model decompositions. The differential inverse kinematics and centroidal momentum models are combined to achieve the wheel velocities which are used in the feedback impedance controller. The double-differential inverse kinematics and centroidal dynamics models are combined to achieve the wheel accelerations which are applied in the feed-forward torque controller. The desired centroidal angular velocity and acceleration are achieved as the byproducts of the wheel motion generator, which are used to track the centroidal momentum and acceleration in the feedback and feed-forward torque controllers. Finally, one concept relating to the prioritized impedance controller is proposed and integrated in the prioritized multi-task torque controller, which is used to track the centroidal motion and validate our wheel motion generator.

The remainder of this paper is organized as follows. In Section 2, the whole-body inverse kinematics is firstly detailed. The wheel motion is extracted out depending on the base motion and the legged motion. Section 3 constitutes the main contribution of this paper which describes the new wheel motion generator to track the robot centroidal motion. The wheel velocities and accelerations are derived in a detailed mathematical way by combining the kinematics model and the centroidal momentum/dynamics model. In Section 4, the robot torque controller is presented by combining our proposed prioritized impedance controller. In Section 5, we use the torque controller to validate our wheel motion generator through numerical simulations using our virtual quadruped-on-wheel robot. Section 6 closes the paper with conclusion and future works.

## 2. System Modeling

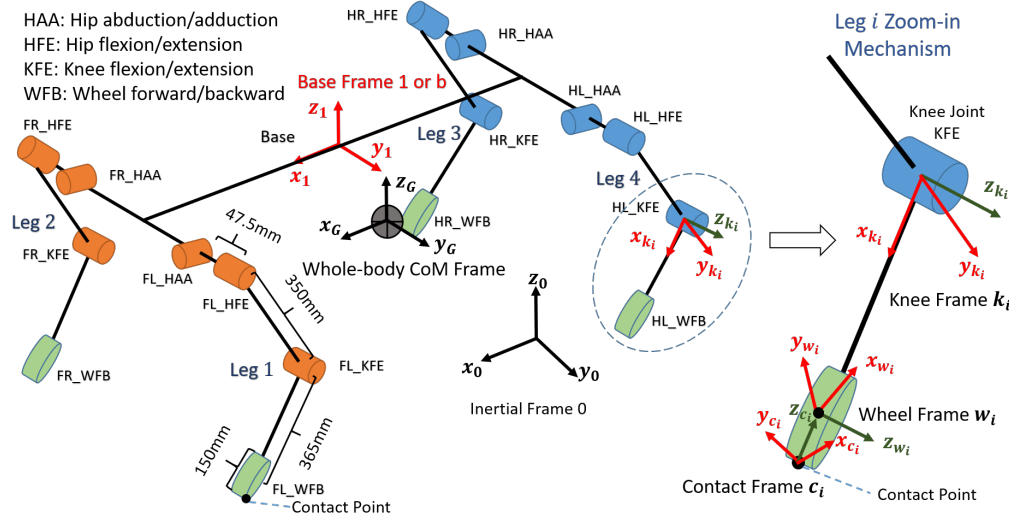
“TowrISIR” is a new quadruped-on-wheel robot that is designed in ISIR laboratory (see Fig. 1), equipped with 12 leg-revolute joints and 4 wheels. The robot mass is about 90kg with length  $\approx 1.06m$ , width  $\approx 0.55m$  and normal height  $\approx 0.8m$ . The leg mass is about 8.6kg and the length parameters are shown in Fig. 2. We select the same hydraulic actuators with HyQ2Max [34] and the power is enough for our robot to handle our designed missions in Section 5. The detailed mechanical design exceeds the topic of this paper. Compared with conventional quadruped robots, the four wheel joints induce more complexity into the kinematics and dynamics models, which will be detailed in this section. In addition, we derive the whole-body dynamics model without any leg massless assumptions. The wheel contact model and the whole-body inverse kinematics are derived as the basis for the wheel motion generator in next section. We concentrate on the rolling mode and all wheels are always maintained in contact with the ground. In this paper, spatial vectors [31] are used and written in upright bold letters, and conventional 3D vectors are written in italic bold letters. For the vectors and matrices in this paper, the subscript  $i \in [1, \dots, 4]$  always denotes the leg index shown in Fig. 2, and  $j \in [1, \dots, 17]$  always represents the rigid body index or the joint index shown in Fig. 3.

### 2.1. Connectivity and Dynamics Model

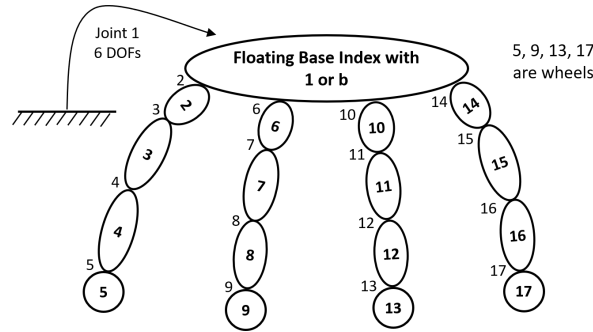
From the connectivity depicted in Fig. 3 and the robot structure in Fig. 2, the joint configuration is defined as,

$$\mathbf{q} = \left[ \mathbf{x}_b^T \quad \mathbf{q}_{g_1}^T \quad \mathbf{q}_{g_2}^T \quad \mathbf{q}_{g_3}^T \quad \mathbf{q}_{g_4}^T \right]^T \in \mathbb{R}^{n_b + 4n_l = n}, \quad (1)$$

where  $\mathbf{x}_b = [\boldsymbol{\theta}_b^T \quad \mathbf{p}_b^T]^T$  is the orientation and position of the floating base in the inertial global coordinate, accounting  $n_b = 6$  degrees of freedom (DOF). The robot has  $n_l = 4$  legs and each leg has 4 joints.  $n = 4n_l + n_b = 22$  denotes the



**Figure 2:** Structure of TowrISIR with leg indexes: The whole-body CoM frame  $G$  is set parallel with the base frame 1 or  $b$ . The frames relating to the inverse kinematics are zoomed in, including the contact frame  $c_i$ , the wheel frame  $w_i$  and the knee frame  $k_i$ . The letters F, L, R and H represent front, left, right and hind. On the front-left (FL) leg, we give the leg lengths. The wheel diameter is 150mm. The contact frame is always estimated at the lowest point in the sagittal plane of the wheel in the gravity direction.



**Figure 3:** Connectivity of TowrISIR with body and joint indexes: the index number inside each ellipse represents the body index, and the outside index numbers denote the joint indexes. Joint 1/b represents the floating-base joint accounting 6 DOFs. We set one frame on each joint, and the frame index has the same index with the joint. There are  $N_B = 17$  bodies, in which the bodies 5, 9, 13 and 17 represent the wheels,  $\mathbf{q}_w = [q_5 \ q_9 \ q_{13} \ q_{17}]^T$ . The base body/frame is the parent body/frame of the bodies/frames 2, 6, 10 and 14. In each leg, the body/frame  $j$  is the parent of the body/frame  $j + 1$ .

whole-body DOF.  $\mathbf{q}_{g_i} = [\mathbf{q}_{r_i}^T \ \mathbf{q}_{w_i}]^T \in \mathbb{R}^4$  represents the leg  $i$  joint configuration, including three leg-revolute joints as  $\mathbf{q}_{r_i} \in \mathbb{R}^3$  and one wheel joint as  $\mathbf{q}_{w_i}$ . We define another two vectors to include each  $\mathbf{q}_{r_i}$  and  $\mathbf{q}_{w_i}$  separately as follows,

$$\mathbf{q}_r = [\mathbf{q}_{r_1}^T \ \mathbf{q}_{r_2}^T \ \mathbf{q}_{r_3}^T \ \mathbf{q}_{r_4}^T]^T \in \mathbb{R}^{n_r=12}, \quad (2a)$$

$$\mathbf{q}_w = [\mathbf{q}_{w_1} \ \mathbf{q}_{w_2} \ \mathbf{q}_{w_3} \ \mathbf{q}_{w_4}]^T \in \mathbb{R}^{n_w=4}, \quad (2b)$$

where  $\mathbf{q}_r$  include all the leg-revolute joints except for the wheels, and  $\mathbf{q}_w$  denotes all wheel-joint configurations. The generalized dynamics model is derived by the Newton-Euler method using the robot structure in Fig. 2 as below,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}, \quad (3)$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{C} \in \mathbb{R}^n$  and  $\mathbf{G} \in \mathbb{R}^n$  denote the generalized inertia matrix, the centripetal & Coriolis force, and the gravity force, respectively. In this paper, all wheels are maintained in contact with the ground.  $\mathbf{J}_c = [\mathbf{J}_{c_1}^T \mathbf{J}_{c_2}^T \mathbf{J}_{c_3}^T \mathbf{J}_{c_4}^T]^T \in \mathbb{R}^{3n_c \times n}$  is the contact Jacobian combining  $n_c = 4$  contact points, and  $\mathbf{J}_{c_i} \in \mathbb{R}^{3 \times n}$ .  $\mathbf{F}_c = [\mathbf{F}_{c_1}^T \mathbf{F}_{c_2}^T \mathbf{F}_{c_3}^T \mathbf{F}_{c_4}^T]^T \in \mathbb{R}^{3n_c}$  represents the four contact forces and  $\mathbf{F}_{c_i} \in \mathbb{R}^3$  is the the leg  $i$  contact force. In addition,  $\boldsymbol{\tau} \in \mathbb{R}^{n_r+n_w}$  denotes the actuated torque and  $\mathbf{S} \in \mathbb{R}^{(n_r+n_w) \times n}$  is one selection matrix.

Assuming that there is no slippage between the wheel contact points and the ground, the velocity of the contact points is expressed by the following constraint in (4a), and its time derivative is given in (4b),

$$\mathbf{v}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (4a)$$

$$\dot{\mathbf{v}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (4b)$$

where  $\mathbf{v}_c = [\mathbf{v}_{c_1}^T \mathbf{v}_{c_2}^T \mathbf{v}_{c_3}^T \mathbf{v}_{c_4}^T]^T$  is the translational velocities of all wheel contact points. Each contact frame  $c_i$  is shown in Fig. 2 and we can see that the  $z$  axis of the contact frame is coincident with the line from the contact point to the wheel center. The wheel contact points of quadruped-on-wheel robots always change, even the wheel joint does not move, e.g. squatting motion. This case is different with the fixed contact points on the legs of conventional quadruped robots [5]. Since the robot knows nothing about the environment, the wheel contact points are estimated at the lowest points of the wheels, which are located in the wheel sagittal plane along the gravity direction.

## 2.2. Whole-body Inverse kinematics

In this subsection, we will decompose the wheel velocities and accelerations out from the contact constraints in (4). Firstly, we re-derive respectively the spatial velocity and acceleration of the leg  $i$  wheel contact point using spatial vectors as follows,

$$\mathbf{v}_{c_i} = \begin{bmatrix} \boldsymbol{\omega}_{c_i} \\ \mathbf{v}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i} - \boldsymbol{\Phi}_{c_i} \mathbf{v}_{w_i[z]}, \quad (5a)$$

$$\dot{\mathbf{v}}_{c_i} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_{c_i} \\ \dot{\mathbf{v}}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_{w_i} \dot{\mathbf{v}}_{w_i} - \boldsymbol{\Phi}_{c_i} \dot{\mathbf{v}}_{w_i[z]} - ({}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i}) \times \boldsymbol{\Phi}_{c_i} \mathbf{v}_{w_i[z]}, \quad (5b)$$

where the symbol  $\times$  denotes the cross product for spatial vectors, detailed in [31].  $\mathbf{v}_{c_i} \in \mathbb{R}^6$  is the spatial velocity of the leg  $i$  contact point.  $\mathbf{v}_{c_i}$  and  $\boldsymbol{\omega}_{c_i}$  denote the translational and angular velocities of the leg  $i$  contact point, respectively.  $\boldsymbol{\Phi}_{c_i} = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$  represents the free modes of the virtual contact joint, which is between the contact point and the ground. The matrix relating to the free modes of one joint should be referred to [28]. The subscript  $w_i$  denotes the leg  $i$  wheel frame, shown in the zoom-in mechanism of Fig. 2.  ${}^{c_i}\mathbf{X}_{w_i}$  is the transformation matrix for spatial vectors from the wheel frame to the contact point frame. All the transformation matrices using the symbol  $\mathbf{X}$  have the size  $6 \times 6$  which are compatible with the spatial vectors [31].  $\mathbf{v}_{w_i}$  is the spatial velocity of the leg  $i$  wheel and  $\mathbf{v}_{w_i[z]}$  is the third component of  $\mathbf{v}_{w_i}$ , namely the wheel angular velocity around the  $z_{w_i}$  axis (see Fig. 2).  $\mathbf{v}_{w_i}$  is the composite result of the legged suspension and the wheel joint motion. Therefore, even when the wheel joint velocity  $\dot{\mathbf{q}}_{w_i}$  is zero,  $\mathbf{v}_{w_i[z]} \neq 0$  for the squatting down motion of the robot. The translational parts of  $\boldsymbol{\Phi}_{c_i} \mathbf{v}_{w_i[z]}$  and  $\boldsymbol{\Phi}_{c_i} \dot{\mathbf{v}}_{w_i[z]}$  are zero and have no influence on the translational motion of the wheel contact points. From the translational components in (5a) and (5b), we can derive the detailed expression of each contact  $\mathbf{J}_{c_i}$  and  $\dot{\mathbf{J}}_c \dot{\mathbf{q}}$ . In addition,  $\mathbf{J}_c$  and  $\dot{\mathbf{J}}_c \dot{\mathbf{q}} = [(\dot{\mathbf{J}}_{c_1} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_2} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_3} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_4} \dot{\mathbf{q}})^T]^T$  are used in the contact-consistent prioritized operational space control [17], shown in Appendix B. All the symbols used in this paper are listed in Tab. 1. The wheel spatial velocity can be derived as below,

$$\mathbf{v}_{w_i} = {}^{w_i}\mathbf{X}_{k_i} \mathbf{v}_{k_i} + \boldsymbol{\Phi}_{w_i} \dot{\mathbf{q}}_{w_i}, \quad (6a)$$

$$\dot{\mathbf{v}}_{w_i} = {}^{w_i}\mathbf{X}_{k_i} \dot{\mathbf{v}}_{k_i} + \boldsymbol{\Phi}_{w_i} \ddot{\mathbf{q}}_{w_i} + \mathbf{v}_{w_i} \times \boldsymbol{\Phi}_{w_i} \dot{\mathbf{q}}_{w_i}, \quad (6b)$$

where the subscript  $k_i$  represents the leg  $i$  knee joint frame, shown in Fig. 2.  ${}^{w_i}\mathbf{X}_{k_i}$  transforms spatial vectors from the knee frame to the wheel frame.  $\mathbf{v}_{k_i}$  is the spatial velocity of the leg  $i$  knee body.  $\boldsymbol{\Phi}_{w_i} = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$  represents the free modes of the leg  $i$  wheel joint. The equation in (6a) is substituted into (5a), then the wheel joint velocity  $\dot{\mathbf{q}}_{w_i}$  can be extracted out by selecting the translational part of (5a) as follows,

$$\mathbf{v}_{c_i} = \mathbf{S}_c \mathbf{v}_{c_i} = \mathbf{S}_c {}^{c_i}\mathbf{X}_{k_i} \mathbf{v}_{k_i} + \mathbf{S}_c {}^{c_i}\mathbf{X}_{w_i} \boldsymbol{\Phi}_{w_i} \dot{\mathbf{q}}_{w_i} = \mathbf{0}, \quad (7)$$



where  $\mathcal{S}_c = [\mathbf{0}_{3 \times 3} \quad \mathbf{1}_{3 \times 3}]$  is a selection matrix.  ${}^{c_i}\mathbf{X}_{k_i}$  transforms spatial vectors from the knee frame to the contact frame. Then we combine all wheel velocities using (7) and extract  $\dot{\mathbf{q}}_w$  out in (8a), and the contact velocity constraint in (4a) is rewritten in (8b),

$$\mathbf{L}_w \dot{\mathbf{q}}_w = \mathbf{L} \begin{bmatrix} \mathbf{v}_b \\ \dot{\mathbf{q}}_r \end{bmatrix}, \quad (8a)$$

$$\mathbf{v}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{L} \begin{bmatrix} \mathbf{v}_b \\ \dot{\mathbf{q}}_r \end{bmatrix} - \mathbf{L}_w \dot{\mathbf{q}}_w = \mathbf{0}, \quad (8b)$$

where  $\mathbf{v}_b = [\boldsymbol{\omega}_b^T \quad \mathbf{v}_b^T]^T$  is the base spatial velocity and is composed by its angular velocity  $\boldsymbol{\omega}_b$  and its translational velocity  $\mathbf{v}_b$ .  $\mathbf{L} \in \mathbb{R}^{3n_c \times (n_b + n_r)}$  is a stack vector relating to the base and the legged joint velocities, and  $\mathbf{L}_w \in \mathbb{R}^{3n_c \times n_w}$  is a block-diagonal matrix relating to the wheel joint velocities as follows,

$$\mathbf{L} = \begin{bmatrix} \mathcal{S}_c {}^{c_1}\mathbf{X}_{k_1} \mathbf{J}_{k_1} \\ \mathcal{S}_c {}^{c_2}\mathbf{X}_{k_2} \mathbf{J}_{k_2} \\ \mathcal{S}_c {}^{c_3}\mathbf{X}_{k_3} \mathbf{J}_{k_3} \\ \mathcal{S}_c {}^{c_4}\mathbf{X}_{k_4} \mathbf{J}_{k_4} \end{bmatrix}, \quad \mathbf{L}_w = \begin{bmatrix} -\left(\mathcal{S}_c {}^{c_1}\mathbf{X}_{w_1} \boldsymbol{\Phi}_{w_1}\right) & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & -\left(\mathcal{S}_c {}^{c_2}\mathbf{X}_{w_2} \boldsymbol{\Phi}_{w_2}\right) & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & -\left(\mathcal{S}_c {}^{c_3}\mathbf{X}_{w_3} \boldsymbol{\Phi}_{w_3}\right) & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & -\left(\mathcal{S}_c {}^{c_4}\mathbf{X}_{w_4} \boldsymbol{\Phi}_{w_4}\right) \end{bmatrix}, \quad (9)$$

where  $\mathbf{0}_{3 \times 1}$  denotes the zero vector.  $\mathbf{J}_{k_i}$  is the Jacobian relating to the knee joint velocity.  $\mathbf{L}$  will be decomposed in (13a) in Section 3.1. In this way, the wheel velocity  $\dot{\mathbf{q}}_w$  is extracted out depending on the base motion and the legged motion in (8a), which serves as the whole-body differential kinematics model for the proposed wheel velocity generator in Section 3.1. For the wheel acceleration,  $\ddot{\mathbf{q}}_{w_i}$  can be extracted out by substituting (6b) into (5b), and we select the translational part of (5b) as below,

$$\dot{\mathbf{v}}_{c_i} = \mathcal{S}_c \dot{\mathbf{v}}_{c_i} = \mathcal{S}_c \left( {}^{c_i}\mathbf{X}_{k_i} \dot{\mathbf{v}}_{k_i} + {}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i} \times \boldsymbol{\Phi}_{w_i} \dot{\mathbf{q}}_{w_i} - {}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i} \times \boldsymbol{\Phi}_{c_i} \mathbf{v}_{w_i[z]} \right) + \mathcal{S}_c {}^{c_i}\mathbf{X}_{w_i} \boldsymbol{\Phi}_{w_i} \ddot{\mathbf{q}}_{w_i} = \mathbf{0}. \quad (10)$$

Similarly, the four wheel accelerations are combined using (10) and extract  $\ddot{\mathbf{q}}_w$  out in (11a), and the contact constraint in (4b) is rewritten in (11b) as follows,

$$\mathbf{L}_w \ddot{\mathbf{q}}_w = \mathbf{L}_a = \begin{bmatrix} \mathcal{S}_c ({}^{c_1}\mathbf{X}_{k_1} \dot{\mathbf{v}}_{k_1} + {}^{c_1}\mathbf{X}_{w_1} \mathbf{v}_{w_1} \times \boldsymbol{\Phi}_{w_1} \dot{\mathbf{q}}_{w_1}) - \mathcal{S}_c ({}^{c_1}\mathbf{X}_{w_1} \mathbf{v}_{w_1}) \times \boldsymbol{\Phi}_{c_1} \mathbf{v}_{w_1[z]} \\ \vdots \\ \mathcal{S}_c ({}^{c_4}\mathbf{X}_{k_4} \dot{\mathbf{v}}_{k_4} + {}^{c_4}\mathbf{X}_{w_4} \mathbf{v}_{w_4} \times \boldsymbol{\Phi}_{w_4} \dot{\mathbf{q}}_{w_4}) - \mathcal{S}_c ({}^{c_4}\mathbf{X}_{w_4} \mathbf{v}_{w_4}) \times \boldsymbol{\Phi}_{c_4} \mathbf{v}_{w_4[z]} \end{bmatrix}, \quad (11a)$$

$$\dot{\mathbf{v}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{L}_a - \mathbf{L}_w \ddot{\mathbf{q}}_w = \mathbf{0}, \quad (11b)$$

where  $\mathbf{L}_a$  is a stack vector and is composed of the items in (10) except items with  $\ddot{\mathbf{q}}_{w_i}$ , and  $\mathbf{L}_a$  will be decomposed in (20a) in Section 3.2. In this way, the wheel acceleration  $\ddot{\mathbf{q}}_w$  is extracted out in (11a), which serves as the whole-body double-differential kinematics model for the proposed wheel acceleration generator in Section 3.2.

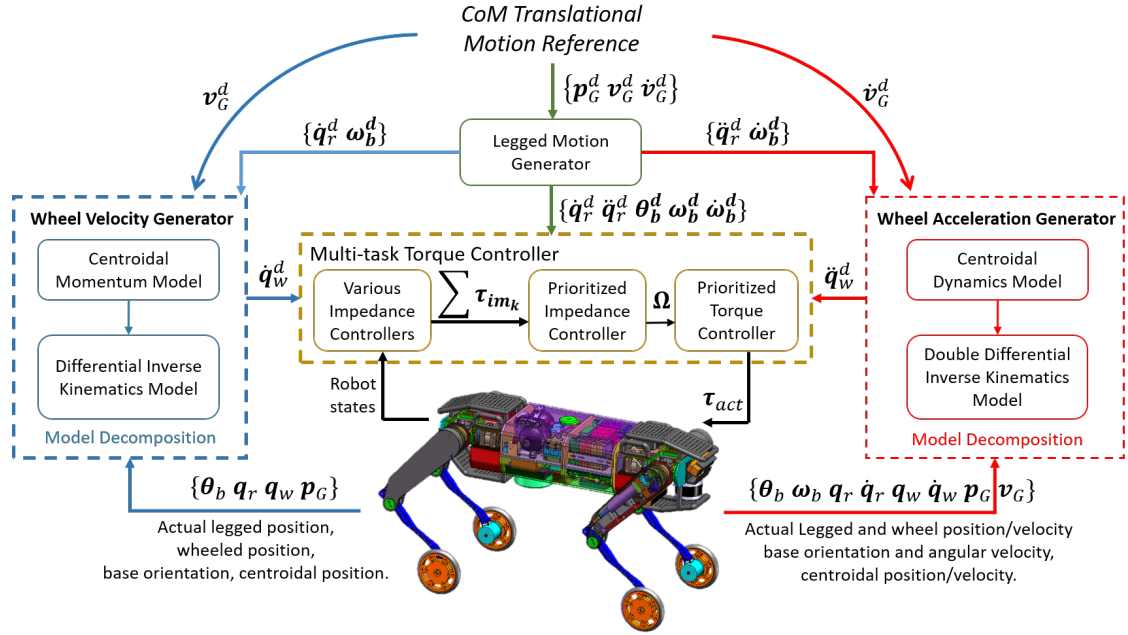
### 2.3. Centroidal Jacobian

To integrate tasks (relating to the robot centroidal motion) into the prioritized operational space torque control framework, we derive its Jacobian  $\mathbf{J}_G \in \mathbb{R}^{6 \times n}$  for the centroidal spatial velocity, and this is one further-forward step in this field which is based on the improved computation method of the centroidal dynamics in [3],

$$\mathbf{J}_G = \mathbf{I}_G^{-1} ({}^b\mathbf{X}_G^T \mathbf{U} \mathbf{M}) = ({}^b\mathbf{X}_G)^{-1} (\mathbf{I}_b^C)^{-1} \mathbf{U} \mathbf{M}, \quad (12)$$

where  $\mathbf{I}_G$  denotes the composite rigid-body inertia at the robot CoM, referred in [28].  $\mathbf{I}_b^C = \mathbf{U} \mathbf{M} \mathbf{U}^T$  represents the base composite inertia and  $\mathbf{U} = [\mathbf{1}_{6 \times 6} \quad \mathbf{0}_{6 \times (n-6)}]$  is a selection matrix.  ${}^b\mathbf{X}_G \in \mathbb{R}^{6 \times 6}$  is the transformation matrix for





**Figure 4:** Wheel Motion Generation Framework: For the rolling locomotion mode, by given the quadruped-CoM translational motion reference, the legged locomotion generator outputs the desired legged configuration for the legged active suspension and the base height control. The superscript  $d$  means “desired”. The wheel velocity and acceleration are derived using the centroidal momentum/dynamics based wheel velocity/acceleration generators in the dashed blue and red boxes, detailed in Section 3. The outputs of the generators serve as the inputs of the prioritized impedance controller based torque controller in the yellow dashed box, detailed in Section 4.

spatial vectors from the robot CoM frame to the base frame whose index is 1 or  $b$ , shown in Fig. 3. The robot CoM frame is shown in Fig. 2.

### 3. Wheel Motion Generator based on Centroidal Momentum and Dynamics

When the robot runs on various rough terrains, tracking the whole-body CoM motion is advantageous since the system translational and angular momentum can be defined at the robot CoM naturally. 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, the quadruped-CoM translational motion serves as the reference. Its forward motion is achieved by the wheeled locomotion, and its vertical and lateral motions are handled by the legged suspension, e.g. the squatting down motion and the motion relating to the lateral push recovery. The desired leg-joint references are generated by one legged motion generator which also outputs the desired base angular motion, shown in Fig. 4, and the superscript  $d$  denotes the desired values of any vector. In this paper, we concentrate on the rolling mode and all legs are maintained in contact with the ground. For the normal stance state of the robot, we simply set the desired legged-joint velocity and acceleration to be zero,  $\dot{q}_r^d = \ddot{q}_r^d = \mathbf{0}_{3n_c \times 1}$ . In addition, the desired base angular velocity and acceleration are also set with zero vectors,  $\omega_b^d = \dot{\omega}_b^d = \mathbf{0}_{3 \times 1}$ . Then the desired velocities and accelerations of the legged motion and the base angular motion are applied as the inputs of the wheel motion generator. Although these settings are simple, the adaptive wheel motion generators (depending on actual legged configuration and velocity) in this section output the adaptive wheel motion reference, then the compliant torque controller in Section 4 enables the robot to handle various rough terrains successfully, shown in Section 5. Since the torque controller is built in the operational space, the desired legged motion setting is not used, detailed in Section 4.

It is known that the wheel contact points always change even when the robot squats down, this is different with the conventional quadruped robots. Therefore, to track the quadruped-CoM motion, the wheel motion should be consistent with the actual legged suspension. The wheel velocities and accelerations are achieved by combining the inverse kinematics model and the centroidal momentum/dynamics model. We use the actual leg configuration to achieve the

desired wheel velocities, and we use the actual leg configuration/velocity to achieve the desired wheel accelerations, which enables the robot to adapt effectively on unknown environment. The inputs of the wheel velocity and acceleration generators are  $\{p_G, v_G^d, \theta_b, \omega_b^d, q_r, \dot{q}_r^d, q_w\}$  and  $\{p_G, v_G, \dot{v}_G^d, \theta_b, \omega_b, \dot{\omega}_b^d, q_r, \dot{q}_r, \ddot{q}_r^d, q_w, \dot{q}_w\}$  respectively (see Fig. 4), and  $p_G$  denotes the quadruped-CoM position. The global solving process: by given the inputs for the wheel velocity and acceleration generators, the base translational velocity  $v_b$  and acceleration  $\dot{v}_b$  can be extracted out from the centroidal momentum and dynamics models respectively. Then  $v_b$  and  $\dot{v}_b$  are substituted into the whole-body inverse kinematics models in (8a) and (11a) respectively. Then the wheel velocity  $\dot{q}_w$  and acceleration  $\ddot{q}_w$  can be derived. The desired wheel position can be estimated in each loop. Several algorithms listed in Tab. 2 are developed for the mathematical derivations of the model decompositions. In the following parts, the superscript of the desired variables  $d$  is omitted. The outputs of the wheel velocity generator are used in the impedance controllers in Section 4.1, and the outputs of the wheel acceleration generator are used in the feed-forward torque controller in Section 4.3.

### 3.1. Wheel Velocity Generator

To track the robot centroidal translational velocity reference, the wheel motion  $\dot{q}_w$  can be derived by combining the differential kinematic model in (13a) and the quadruped centroidal momentum model in (13b) as follows,

$$L_w \dot{q}_w = L \begin{bmatrix} v_b \\ \dot{q}_r \end{bmatrix} = L_{b_\omega} \omega_b + L_{b_v} v_b + L_r \dot{q}_r, \quad (13a)$$

$$\mathbf{h}_G = \begin{bmatrix} k_G \\ l_G \end{bmatrix} = \mathbf{I}_G v_G = [\mathbf{I}_{G_\omega} \quad \mathbf{I}_{G_v}] \begin{bmatrix} \omega_G \\ v_G \end{bmatrix} = {}^b \mathbf{X}_G^T \sum_{j=1}^{N_B} {}^j \mathbf{X}_b^T \mathbf{I}_j v_j, \quad (13b)$$

where (13a) is equivalent with (8a) by setting  $L = [L_{b_\omega} \quad L_{b_v} \quad L_r]$ , in which  $L_{b_\omega}$  and  $L_{b_v}$  are the first and second three columns of  $L$  in (9) respectively.  $L_r \dot{q}_r$  is one part contributing to the wheel contact point velocity in (8b), which is composed by the leg-revolute joints (except the wheel joints), and it can be easily achieved using the algorithm in Tab. 2(a).  $\mathbf{h}_G$  denotes the centroidal momentum and it is composed by the angular momentum  $k_G$  and the translational momentum  $l_G$ . In addition,  $v_G$  represents the centroidal spatial velocity,  $\omega_G$  and  $v_G$  represent the centroidal angular and translational velocity respectively. It is noticed that the spatial acceleration is defined as the rate of change of the spatial velocity, and it differs from the classical textbook definition of the rigid-body acceleration, referred in [28]. Therefore, we have the relationship between  $\dot{p}_G$  and  $\dot{v}_G$ , as well as  $\dot{p}_b$  and  $\dot{v}_b$  detailed in Appendix A.  $\mathbf{I}_{G_\omega}$  and  $\mathbf{I}_{G_v}$  are the left and right three columns of  $\mathbf{I}_G$ .  ${}^j \mathbf{X}_b$  transforms spatial vectors from the base frame to the body  $j$  frame.  $\mathbf{I}_j \in \mathbb{R}^{6 \times 6}$  and  $v_j$  represent the body  $j$  spatial inertia and spatial velocity respectively. The body and joint indexes are shown in Fig. 3.

The centroidal momentum model in (13b) can be decomposed into components relating to the base motion, the legged motion and the wheeled motion as follows,

$$\mathbf{I}_G \begin{bmatrix} \omega_G \\ v_G \end{bmatrix} = \mathbf{Q}_b \begin{bmatrix} \omega_b \\ v_b \end{bmatrix} + \mathbf{Q}_r \dot{q}_r + \mathbf{Q}_w \dot{q}_w, \quad (14)$$

where  $\mathbf{Q}_r \dot{q}_r \in \mathbb{R}^6$  denotes one component of the centroidal momentum which is contributed by the leg bodies (except the wheels), and it can be derived using the algorithm in Tab. 2(b).  $\mathbf{Q}_b \in \mathbb{R}^{6 \times 6}$  and  $\mathbf{Q}_w \in \mathbb{R}^{6 \times 4}$  are inertia-like matrices relating to the base spatial motion and the wheel motion as below,

$$\mathbf{Q}_b = [\mathbf{Q}_{b_\omega} \quad \mathbf{Q}_{b_v}] = {}^b \mathbf{X}_G^T \mathbf{I}_b^C, \quad (15a)$$

$$\mathbf{Q}_w = {}^b \mathbf{X}_G^T [{}^{w_1} \mathbf{X}_b^T \mathbf{I}_{w_1} \Phi_{w_1} \quad \dots \quad {}^{w_4} \mathbf{X}_b^T \mathbf{I}_{w_4} \Phi_{w_4}], \quad (15b)$$

where  $\mathbf{Q}_{b_\omega}$  and  $\mathbf{Q}_{b_v}$  are the left and right three columns of  $\mathbf{Q}_b$ . In addition,  $\mathbf{I}_{w_i}$  is the leg  $i$  wheel inertia, and  ${}^{w_i} \mathbf{X}_b$  transforms spatial vectors from the base frame to the leg  $i$  wheel frame. To achieve the wheel velocities, firstly we extract the base translational velocity out mathematically from (14) as follows,

$$v_b = \Gamma_{Gb} \begin{bmatrix} \omega_b \\ v_G \end{bmatrix} - \Gamma_r \dot{q}_r - \Gamma_w \dot{q}_w, \quad (16)$$

where  $\Gamma_{Gb}$ ,  $\Gamma_r \dot{q}_r$  and  $\Gamma_w$  are derived as below,

$$\Gamma_{Gb} = S_c \begin{bmatrix} -\mathbf{I}_{G_\omega} & \mathbf{Q}_{b_v} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{Q}_{b_\omega} & \mathbf{I}_{G_v} \end{bmatrix} \in \mathbb{R}^{3 \times 6}, \quad (17a)$$

$$\Gamma_r \dot{q}_r = S_c \begin{bmatrix} -\mathbf{I}_{G_\omega} & \mathbf{Q}_{b_v} \end{bmatrix}^{-1} \mathbf{Q}_r \dot{q}_r \in \mathbb{R}^3, \quad (17b)$$

$$\Gamma_w = S_c \begin{bmatrix} -\mathbf{I}_{G_\omega} & \mathbf{Q}_{b_v} \end{bmatrix}^{-1} \mathbf{Q}_w \in \mathbb{R}^{3 \times 4}. \quad (17c)$$

In this way, we achieve the relationship between the base translational velocity and the wheel velocity in (16), and it is substituted into the differential kinematics model in (13a),

$$\left( \mathbf{L}_w + \mathbf{L}_{b_v} \Gamma_w \right) \dot{q}_w = \mathbf{L}_{b_v} \left( \Gamma_{Gb} \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_G \end{bmatrix} - \Gamma_r \dot{q}_r \right) + \mathbf{L}_r \dot{q}_r + \mathbf{L}_{b_\omega} \boldsymbol{\omega}_b, \quad (18)$$

where  $\dot{q}_w$  can be derived using the Moore-Penrose inverse method.

After achieving the wheel velocity  $\dot{q}_w$ , the robot centroidal angular velocity  $\boldsymbol{\omega}_G$  and the base translational velocity  $\mathbf{v}_b$  can also be derived as by-products as follows,

$$\begin{bmatrix} \boldsymbol{\omega}_G \\ \mathbf{v}_b \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{G_\omega} & -\mathbf{Q}_{b_v} \end{bmatrix}^{-1} \left( \begin{bmatrix} \mathbf{Q}_{b_\omega} & -\mathbf{I}_{G_v} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_G \end{bmatrix} + \mathbf{Q}_r \dot{q}_r + \mathbf{Q}_w \dot{q}_w \right). \quad (19)$$

Then all the velocity references will be used for the whole-body compliance control in Section 4.1.

### 3.2. Wheel Acceleration Generator

To track the robot centroidal acceleration reference, the wheel acceleration  $\ddot{q}_w$  can be derived using the double-differential kinematic model and the centroidal dynamics model as follows,

$$\mathbf{L}_w \ddot{q}_w = \mathbf{L}_{b_\omega} \dot{\boldsymbol{\omega}}_b + \mathbf{L}_{b_v} \dot{\mathbf{v}}_b + \mathbf{L}_d = \mathbf{L}_a, \quad (20a)$$

$$\dot{\mathbf{h}}_G = \begin{bmatrix} \dot{k}_G \\ \dot{l}_G \end{bmatrix} = \mathbf{I}_G \dot{\mathbf{v}}_G + \mathbf{v}_G \times \mathbf{I}_G \mathbf{v}_G = {}^b \mathbf{X}_G^T \sum_{j=1}^{N_B} j \mathbf{X}_b^T (\mathbf{I}_j \mathbf{a}_j + \mathbf{v}_j \times \mathbf{I}_j \mathbf{v}_j), \quad (20b)$$

where  $\mathbf{a}_j$  is the body  $j$  spatial acceleration. (20a) is equivalent with (11a) by dividing  $\mathbf{L}_a$  into components relating to the base angular and translational motions, and  $\mathbf{L}_d \in \mathbb{R}^{3n_c}$  includes the other items of  $\mathbf{L}_a$ . In addition,  $\mathbf{L}_d$  can be calculated using the algorithm in Tab. 2(c).

Then we decompose the centroidal dynamics model in (20b) into components relating to the base motion, the legged motion, the wheel motion and one bias item as below,

$$\mathbf{I}_G \begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \dot{\mathbf{v}}_G \end{bmatrix} = \mathbf{Q}_b \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_b \end{bmatrix} + \mathbf{Q}_r \ddot{q}_r + \mathbf{Q}_w \ddot{q}_w - \mathbf{Q}_{bias}, \quad (21)$$

where  $\mathbf{Q}_r \ddot{q}_r \in \mathbb{R}^6$  and  $\mathbf{Q}_{bias} \in \mathbb{R}^6$  can be calculated recursively using the algorithm in Tab. 2(d).

To achieve the wheel accelerations, firstly we extract the base translational acceleration out from (21) as below,

$$\dot{\mathbf{v}}_b = \Gamma_{Gb} \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} - \Gamma_w \ddot{q}_w + \Gamma_d, \quad (22)$$

where  $\Gamma_{Gb}$  and  $\Gamma_w$  are referred to (17), and  $\Gamma_d \in \mathbb{R}^3$  is derived as follows,

$$\Gamma_d = \Gamma_{bias} - \Gamma_r \ddot{q}_r, \quad (23a)$$

$$\Gamma_{bias} = S_c \begin{bmatrix} -\mathbf{I}_{G_\omega} & \mathbf{Q}_{b_v} \end{bmatrix}^{-1} \mathbf{Q}_{bias}, \quad (23b)$$

$$\Gamma_r \ddot{q}_r = S_c \begin{bmatrix} -\mathbf{I}_{G_\omega} & \mathbf{Q}_{b_v} \end{bmatrix}^{-1} \mathbf{Q}_r \ddot{q}_r. \quad (23c)$$

**Table 2**

Algorithms for the Wheel Motion Generation:  $p(j)$  denotes the parent body/frame of the body/frame  $j$ . The parent relationship is described in the caption of Fig. 3 relating to the robot connectivity.  ${}^j\mathbf{X}_{p(j)} \in \mathbb{R}^{6 \times 6}$  is the transformation matrix for spatial vectors from the parent frame  $p(j)$  to the frame  $j$ .  $\Phi_j$  represents the free modes of the joint  $j$ ,  $\Phi_1 = \mathbf{1}_{6 \times 6}$  for the floating base joint and  $\Phi_j = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$  for the actuated joints. In the algorithms (a) and (b) relating to the wheel velocity generator, we assume that the velocities are zero only for the floating-base and the wheel joints. In the algorithms (c) and (d) relating to the wheel acceleration generator, we assume that the accelerations are zero only for the floating-base and the wheels joints, and all joint/body velocities are with the real values. In addition, all transformation matrices and each spatial body inertia  $\mathbf{I}_j$  in all algorithms have been calculated with the actual values.

---

(a) Algorithm for  $\mathbf{L}_r \dot{\mathbf{q}}_r$  in (13a) for the wheel velocity generator.

---

Assumption: The velocities of the floating-base joint and wheel joints are zero:  $\mathbf{v}_b = \mathbf{v}_1 = \mathbf{0}$ ,  $\dot{\mathbf{q}}_w = \mathbf{0}$ .

$$\mathbf{L}_r \dot{\mathbf{q}}_r = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{J}_c \begin{bmatrix} \mathbf{0}_{1 \times 6} & \dot{\mathbf{q}}_{r_1}^T & 0 & \dot{\mathbf{q}}_{r_2}^T & 0 & \dot{\mathbf{q}}_{r_3}^T & 0 & \dot{\mathbf{q}}_{r_4}^T & 0 \end{bmatrix}^T$$


---

(b) Algorithm for  $\mathbf{Q}_r \dot{\mathbf{q}}_r$  in (14) for the wheel velocity generator.

---

Assumption: the velocities of the floating-base and wheel joints are zero:  $\mathbf{v}_b = \mathbf{v}_1 = \mathbf{0}$ ,  $\dot{\mathbf{q}}_w = [\dot{q}_5 \ \dot{q}_9 \ \dot{q}_{13} \ \dot{q}_{17}]^T = \mathbf{0}$ .

Initialize one middle variable  $\mathbf{E} = \mathbf{0}$ .

**for**  $j=2$  to  $N_B$  **do**

$$\mathbf{v}_j = {}^j\mathbf{X}_{p(j)} \mathbf{v}_{p(j)} + \Phi_j \dot{\mathbf{q}}_j$$

$$\mathbf{E} = \mathbf{E} + {}^j\mathbf{X}_b^T \mathbf{I}_j \mathbf{v}_j$$

**end**

$$\mathbf{Q}_r \dot{\mathbf{q}}_r = {}^b\mathbf{X}_G^T \mathbf{E}$$


---

(c) Algorithm for  $\mathbf{L}_d$  in (20a) for the wheel acceleration generator.

---

Assumption: The accelerations of the floating-base and wheel joints are zero:  $\dot{\mathbf{v}}_b = \dot{\mathbf{v}}_1 = \mathbf{0}$ ,  $\ddot{\mathbf{q}}_w = [\ddot{q}_5 \ \ddot{q}_9 \ \ddot{q}_{13} \ \ddot{q}_{17}]^T = \mathbf{0}$ .

**for**  $j=2$  to  $N_B$  **do**

$$\dot{\mathbf{v}}_j = {}^j\mathbf{X}_{p(j)} \dot{\mathbf{v}}_{p(j)} + \Phi_j \ddot{\mathbf{q}}_j + \mathbf{v}_j \times \Phi_j \dot{\mathbf{q}}_j$$

**end**

$$\mathbf{L}_d = \begin{bmatrix} \mathbf{S}_c \left( {}^{c_1}\mathbf{X}_{w_1} \dot{\mathbf{v}}_5 - ({}^{c_1}\mathbf{X}_{w_1} \mathbf{v}_{w_1}) \times \Phi_{c_1} \mathbf{v}_{w_1[z]} \right) \\ \mathbf{S}_c \left( {}^{c_2}\mathbf{X}_{w_2} \dot{\mathbf{v}}_9 - ({}^{c_2}\mathbf{X}_{w_2} \mathbf{v}_{w_2}) \times \Phi_{c_2} \mathbf{v}_{w_2[z]} \right) \\ \mathbf{S}_c \left( {}^{c_3}\mathbf{X}_{w_3} \dot{\mathbf{v}}_{13} - ({}^{c_3}\mathbf{X}_{w_3} \mathbf{v}_{w_3}) \times \Phi_{c_3} \mathbf{v}_{w_3[z]} \right) \\ \mathbf{S}_c \left( {}^{c_4}\mathbf{X}_{w_4} \dot{\mathbf{v}}_{17} - ({}^{c_4}\mathbf{X}_{w_4} \mathbf{v}_{w_4}) \times \Phi_{c_4} \mathbf{v}_{w_4[z]} \right) \end{bmatrix}$$


---

(d) Algorithm for  $\mathbf{Q}_r \ddot{\mathbf{q}}_r$  and  $\mathbf{Q}_{bias}$  in (21) for the wheel acceleration generator.

---

Define  $\mathbf{B}$ ,  $\mathbf{E}$ ,  $\mathbf{e}_j$  and  $\mathbf{b}_j$  as middle-process variables.

Assumption: The accelerations of the floating-base and wheel joints are zero:  $\mathbf{a}_b = \mathbf{a}_1 = \mathbf{0}$ ,  $\ddot{\mathbf{q}}_w = [\ddot{q}_5 \ \ddot{q}_9 \ \ddot{q}_{13} \ \ddot{q}_{17}]^T = \mathbf{0}$ .

Initialize  $\mathbf{B} = \mathbf{v}_b \times \mathbf{I}_b \mathbf{v}_b = \mathbf{v}_1 \times \mathbf{I}_1 \mathbf{v}_1$ ,  $\mathbf{E} = \mathbf{0}$ ,  $\mathbf{e}_1 = \mathbf{0}$ ,  $\mathbf{b}_1 = \mathbf{0}$ .

**for**  $j=2$  to  $N_B$  **do**

$$\mathbf{e}_j = {}^j\mathbf{X}_{p(j)} \mathbf{e}_{p(j)} + \Phi_j \ddot{\mathbf{q}}_j$$

$$\mathbf{E} = \mathbf{E} + {}^j\mathbf{X}_1^T \mathbf{I}_j \mathbf{e}_j$$

$$\mathbf{b}_j = {}^j\mathbf{X}_{p(j)} \mathbf{b}_{p(j)} + \mathbf{v}_j \times \Phi_j \dot{\mathbf{q}}_j$$

$$\mathbf{B} = \mathbf{B} + {}^j\mathbf{X}_1^T (\mathbf{I}_j \mathbf{b}_j + \mathbf{v}_j \times \mathbf{I}_j \mathbf{v}_j)$$

**end**

$$\mathbf{Q}_r \ddot{\mathbf{q}}_r = {}^1\mathbf{X}_G^T \mathbf{E} \text{ and } \mathbf{Q}_{bias} = \mathbf{v}_G \times \mathbf{I}_G \mathbf{v}_G - {}^1\mathbf{X}_G^T \mathbf{B}$$


---

In this way, we have achieved the relationship between the base translational acceleration and the wheel acceleration in (22). Then we substitute this relationship function (22) into the double-differential kinematics model (20a),

$$\left( \mathbf{L}_w + \mathbf{L}_{b_v} \Gamma_w \right) \ddot{\mathbf{q}}_w = \mathbf{L}_{b_v} \left( \Gamma_{Gb} \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} + \Gamma_d \right) + \mathbf{L}_d + \mathbf{L}_{b_w} \dot{\boldsymbol{\omega}}_b, \quad (24)$$

where  $\ddot{\mathbf{q}}_w$  can be derived using the Moore-Penrose inverse method. After achieving the wheel acceleration, the base translational acceleration  $\dot{\mathbf{v}}_b$  and the robot centroidal angular acceleration  $\dot{\boldsymbol{\omega}}_G$  can be derived as by-products as below,

$$\begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \dot{\mathbf{v}}_b \end{bmatrix} = [\mathbf{I}_{G_\omega} \quad -\mathbf{Q}_{b_v}]^{-1} \left( [\mathbf{Q}_{b_\omega} \quad -\mathbf{I}_{G_v}] \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} + \mathbf{Q}_r \ddot{\mathbf{q}}_r + \mathbf{Q}_w \ddot{\mathbf{q}}_w - \mathbf{Q}_{bias} \right). \quad (25)$$

All the acceleration references have been achieved and will be used in the feed-forward torque controller in Section 4.3. In addition, the wheel position can be estimated in each control loop as follows,

$$\mathbf{q}_{w|k+1} = \mathbf{q}_{w|k} + \delta t \dot{\mathbf{q}}_{w|k} + \frac{1}{2} \delta t^2 \ddot{\mathbf{q}}_{w|k}, \quad (26)$$

where  $\mathbf{q}_{w|0}$  applies the initial wheel position.  $\delta t$  represents one calculation loop time period. The subscript  $|k$  denotes the  $k^{th}$  loop value. The wheel position reference is used in the impedance controller in Section 4.1.

In this section, we achieve the desired wheel motion by further combining the centroidal momentum and dynamics models, and this method can be used for other locomotion robots with wheels. In addition, the online wheel motion generator enables the robot with more robustness due to the fact that the velocity is achieved using the actual robot configuration, and the acceleration is derived by the robot actual configuration and the actual body velocities. This strategy can also be applied for online motion generators of other robots without wheels.

## 4. Whole-body Torque Controller

In this section, we define several impedance controllers to enable the robot with compliance abilities. Then we propose one concept, ‘‘prioritized impedance controller’’, which enables various impedance forces to be consistent with the multi-task operational space control framework. The desired positions and velocities are applied in the compliance controllers. The desired accelerations are applied in the feed-forward torque controller.

For the rolling based locomotion on rough terrain, we control three tasks which are ranked in priority order from high to low as: the contact constraint task (signed as  $c$ ) accounting for 12 DOFs, the centroidal motion tracking task (signed as  $G$ ) accounting for 6 DOFs and the wheel motion tracking task (signed as  $w$ ) accounting for 4 DOFs. Therefore, the sum number of these three-task DOFs is equal to the whole-body DOF  $n = 22$ .

### 4.1. Feedback Whole-body Impedance Controllers

When the robot runs in the unknown environment, the uncertainties include the unknown terrain, the imprecise contact points, the unknown rolling resistance and the external force. Therefore, several whole-body impedance controllers are required to be built to compensate the influence to our kinematics and dynamics models, and enable the robot to run on rough terrain with compliance. In this subsection, all impedance controllers are established in the task space and they are projected into the generalized coordinate space.

Initially, we define one impedance controller for the whole-body angular and translational momentums to track the robot centroidal motion. Since the angular position of the whole-body CoM has no sense, we apply only the damping matrix  $\mathbf{D}_{G_\omega}$  for the angular momentum tracking as below,

$$\boldsymbol{\tau}_{im_G} = \mathbf{J}_G^T \left[ \mathbf{K}_{G_v} (\mathbf{p}_G^d - \mathbf{p}_G) + \mathbf{D}_{G_\omega} (\mathbf{k}_G^d - \mathbf{k}_G) + \mathbf{D}_{G_v} (\mathbf{v}_G^d - \mathbf{v}_G) \right] \in \mathbb{R}^n, \quad (27)$$

where  $\mathbf{K}_{G_v}$  and  $\mathbf{D}_{G_v}$  denote the stiffness and damping matrices whose subscripts  $G_v$  represent the robot centroidal translational motion. In this section, all the superscript  $d$  means the desired values of any vector. In addition, one impedance controller is designed to enable the base with small angular motions as follows,

$$\boldsymbol{\tau}_{im_b} = \mathbf{J}_b^T \left[ \mathbf{K}_b (\boldsymbol{\theta}_b^d - \boldsymbol{\theta}_b) + \mathbf{D}_b (\boldsymbol{\omega}_b^d - \boldsymbol{\omega}_b) \right] \in \mathbb{R}^n, \quad (28)$$

where  $\mathbf{J}_b \in \mathbb{R}^{3 \times n}$  is the Jacobian for the base angular motion.  $\mathbf{K}_b$  and  $\mathbf{D}_b$  denote respectively the relative angular stiffness and damping matrices. Furthermore, the active compliances for the legged suspension and the wheel motion play a significant role for the robot to cross over various rough terrains. Therefore, we establish two impedance controllers

on all leg-end points and all wheel joints respectively as below,

$$\boldsymbol{\tau}_{im_p} = \mathbf{J}_p^T [\mathbf{K}_p (\tilde{\mathbf{p}}_w^d - \tilde{\mathbf{p}}_w) + \mathbf{D}_p (\dot{\tilde{\mathbf{p}}}_w^d - \dot{\tilde{\mathbf{p}}}_w)] \in \mathbb{R}^n, \quad (29a)$$

$$\boldsymbol{\tau}_{im_w} = \mathbf{J}_w^T [\mathbf{K}_w (\mathbf{q}_w^d - \mathbf{q}_w) + \mathbf{D}_w (\dot{\mathbf{q}}_w^d - \dot{\mathbf{q}}_w)] \in \mathbb{R}^n, \quad (29b)$$

where  $\tilde{\mathbf{p}}_w \in \mathbb{R}^{3n_c}$  combines all the wheel-center positions relative to the base frame.  $\mathbf{J}_p \in \mathbb{R}^{3n_c \times n}$  denotes the Jacobian for all the wheel-center motion relative to the base frame.  $\mathbf{J}_w \in \mathbb{R}^{n_c \times n}$  represents the Jacobian relating to all wheel-joint motion.  $\mathbf{K}_p$ ,  $\mathbf{D}_p$ ,  $\mathbf{K}_w$  and  $\mathbf{D}_w$  denote the stiffnesses and dampings for the legged suspension and the wheel motion respectively.

To enable the robot to handle various terrains, we set different stiffnesses and dampings for the centroidal translational motion in the vertical direction in (27) and the wheel-center motion relative to the base in (29a). On the terrain with discontinuous stones, stairs and continuous bumps without much altitude variation, we concern more importance on the height control of the centroidal translational motion, and the components of the stiffness and damping for the wheel-center motion in the base vertical direction are set with small values. On the terrain with complex slopes, the stiffness and damping for the centroidal height control is set to be low and we give more concern on the wheel-center motion relative to the base.

In this part, we define four impedance controllers in this subsection, in which (27) and (29b) are related to the designed tasks, (28) and (29a) are general impedance controllers indirectly influencing on the designed tasks.

## 4.2. Prioritized Impedance Controller

We propose the prioritized impedance controller, which means that all impedance controllers should be consistent with the task hierarchy. We contribute to prioritize various impedance torques in Section 4.1 and combine them into one single torque  $\boldsymbol{\Omega} \in \mathbb{R}^n$  in the generalized coordinate space (cf. Fig. 4) as follows,

$$\boldsymbol{\Omega} = \left[ \mathbf{J}_c^T \bar{\mathbf{J}}_c^T + \mathbf{J}_{G|c}^T \bar{\mathbf{J}}_{G|c}^T + \mathbf{J}_{w|G|c}^T \bar{\mathbf{J}}_{w|G|c}^T + \mathbf{N}_{all} \right] \sum \boldsymbol{\tau}_{im_k}, \quad (30)$$

where  $\sum \boldsymbol{\tau}_{im_k} = \boldsymbol{\tau}_{im_G} + \boldsymbol{\tau}_{im_b} + \boldsymbol{\tau}_{im_g} + \boldsymbol{\tau}_{im_w}$  sums all the impedance torques in Section 4.1.  $\sum \boldsymbol{\tau}_{im_k}$  can also integrate any other user-defined impedance controllers.  $\mathbf{J}_{G|c} = \mathbf{J}_G \mathbf{N}_c$  and  $\mathbf{J}_{w|G|c} = \mathbf{J}_w \mathbf{N}_{G|c}$  represent the Jacobians for the centroidal motion and the wheel motion tasks respectively.  $\mathbf{N}_c = \mathbf{1}_{n \times n} - \bar{\mathbf{J}}_c \mathbf{J}_c$  denotes the null-space projector for the contact constraint task, and  $\mathbf{N}_{G|c} = \mathbf{N}_c - \bar{\mathbf{J}}_{G|c} \mathbf{J}_{G|c}$  is the null-space projector combining the centroidal motion and the contact constraint tasks. The subscript  $G|c$  means that the CoM tracking task cannot conflict the contact constraint and should be in the null-space of the contact-constraint task. Similarly, the subscript  $w|G|c$  means that the wheel motion task should be in the null-space of the contact and CoM tracking tasks. Each  $\bar{\mathbf{J}}$  denotes the dynamically-consistent inverse of each task Jacobian  $\mathbf{J}$ , detailed in Appendix B. In addition,  $\mathbf{N}_{all} = \mathbf{N}_{G|c} - \bar{\mathbf{J}}_{w|G|c} \mathbf{J}_{w|G|c}$  is the null-space projector combining null-spaces of the three designed tasks and  $\mathbf{N}_{all} = \mathbf{0}_{n \times n}$  in this paper because all robot DOFs are taken into account. The detailed null-space based inverse dynamics should be referred to [16], and some of the relative properties are listed in Appendix C. The impedance forces in the operational space for the three hierarchal tasks have the following relationships,

$$\bar{\mathbf{J}}_c^T \boldsymbol{\Omega} = \bar{\mathbf{J}}_c^T \sum \boldsymbol{\tau}_{im_k}, \quad \text{for the contact constraint task with } \bar{\mathbf{J}}_c^T \mathbf{J}_{G|c}^T = \mathbf{0} \text{ and } \bar{\mathbf{J}}_c^T \mathbf{J}_{w|G|c}^T = \mathbf{0}, \quad (31a)$$

$$\bar{\mathbf{J}}_{G|c}^T \boldsymbol{\Omega} = \bar{\mathbf{J}}_{G|c}^T \sum \boldsymbol{\tau}_{im_k}, \quad \text{for the centroidal motion task with } \bar{\mathbf{J}}_{G|c}^T \mathbf{J}_c^T = \mathbf{0} \text{ and } \bar{\mathbf{J}}_{G|c}^T \mathbf{J}_{w|G|c}^T = \mathbf{0}, \quad (31b)$$

$$\bar{\mathbf{J}}_{w|G|c}^T \boldsymbol{\Omega} = \bar{\mathbf{J}}_{w|G|c}^T \sum \boldsymbol{\tau}_{im_k}, \quad \text{for the wheel motion task with } \bar{\mathbf{J}}_{w|G|c}^T \mathbf{J}_c^T = \mathbf{0} \text{ and } \bar{\mathbf{J}}_{w|G|c}^T \mathbf{J}_{G|c}^T = \mathbf{0}, \quad (31c)$$

where we use the properties in Appendix C. The relationships in (31) show that the impedance force in the operational space for one task is decoupled from all the other tasks. Furthermore, the prioritized impedance controller in (30) can integrate the impedance controllers relating to the designed tasks, e.g. including the centroidal motion task in (27) and the wheel motion task in (29b). Especially, compared with the conventional impedance controllers, it has three more advantages:

- Several general impedance controllers indirectly acting on the designed tasks, e.g. the impedance controllers in (28) and (29a), can also be integrated into the single torque in (30).

- By using the null-space based projections, the prioritized impedance controller satisfies the task hierarchies and influences the designed tasks in a decoupled way.
- The prioritized impedance controller can also be integrated into one optimization based torque controller, e.g. multi-task hierarchical convex optimization using the quadratic programming (QP) technique, which combines the above two advantages and is able to handle inequality constraints.

For robots with higher DOFs, the designed tasks may not take into account all the robot DOFs,  $\mathbf{N}_{all}$  in (30) is not a zero matrix which enables the robot with abilities to handle other impedance controllers without conflicting the task hierarchy. The prioritized impedance controller is applied to achieve the feedback torque  $\boldsymbol{\tau}_{fb} \in \mathbb{R}^{n_r+n_w}$  as below,

$$\boldsymbol{\tau}_{fb} = \mathbf{P}\boldsymbol{\Omega} = \left[ (\mathbf{P}_{QR}\mathbf{S}^T)^+ \mathbf{P}_{QR} \right] \boldsymbol{\Omega}, \quad (32)$$

where  $\mathbf{P} \in \mathbb{R}^{(n_r+n_w) \times n}$  denotes one mapping matrix and is derived by the QR decomposition projector of the contact Jacobian  $\mathbf{J}_c$ , which enables the torque for the actuators to be independent of the contact force [33], i.e.  $\mathbf{P}\mathbf{J}_c^T\mathbf{F}_c = \mathbf{0}$ . The superscript  $+$  denotes the Moore-Penrose inverse. The proposed impedance controller in (30) and the feedback force in (32) both respect the contact constraints, which is consistent with the torque controller in Section 4.3.

In this subsection, the proposed prioritized impedance controller can be used for other robots with high DOFs. For example, when an obstacle is approaching the head of one humanoid robot in [16], the reactive force generated by the repulsive potential field pushes the robot to avoid collision. In the process, it is preferable to avoid the obstacle by using the reactive whole-body motion (e.g. the centroidal motion), not only the head motion. In this sense, the proposed impedance controller enables this reactive force to influence the whole-body behaviors by respecting task hierarchies. In addition, the method can be integrated into various hierarchical multi-task torque controllers, including the pseudo-inverse based dynamic method in Section 4.3, and other torque controllers using optimization based techniques which is detailed in our work [35](submitted).

### 4.3. Prioritized Torque Controller

The whole-body feed-forward torque controller applies the contact-consistent hierarchical control framework [17]. In this framework, the operational space force for the contact constraint task is not required to be calculated, shown in the equation (35). For the other two tasks, we define  $\mathbf{F}_{G|c}$  and  $\mathbf{F}_{w|G|c}$  to represent the operational space forces for the centroidal motion task and the wheel motion task respectively, and they are consistent with the contact constraint. For the centroidal motion task,  $\mathbf{F}_{G|c}$  can be achieved by given the task-acceleration reference  $\dot{\mathbf{v}}_G^d$  as follows,

$$\mathbf{F}_{G|c} = \boldsymbol{\Lambda}_{G|c}\dot{\mathbf{v}}_G^d + \boldsymbol{\mu}_{G|c} + \boldsymbol{\rho}_{G|c}, \quad (33)$$

where  $\boldsymbol{\Lambda}_{G|c}$ ,  $\boldsymbol{\mu}_{G|c}$  and  $\boldsymbol{\rho}_{G|c}$  denote the operational space inertia, the Coriolis and centrifugal force, and gravity force for the centroidal motion task. To avoid incompleteness, we list these variables in Appendix B. For the wheel motion task,  $\mathbf{F}_{w|G|c}$  will not conflict the centroidal motion task, and can be achieved by given the task-acceleration reference  $\ddot{\mathbf{q}}_w^d$  as below,

$$\mathbf{F}_{w|G|c} = \boldsymbol{\Lambda}_{w|G|c}\ddot{\mathbf{q}}_w^d + \boldsymbol{\mu}_{w|G|c} + \boldsymbol{\rho}_{w|G|c} - \boldsymbol{\Lambda}_{w|G|c}\mathbf{J}_w\mathbf{M}^{-1}\left(\mathbf{J}_{G|c}^T\mathbf{F}_{G|c}\right), \quad (34)$$

where  $\boldsymbol{\Lambda}_{w|G|c}$ ,  $\boldsymbol{\mu}_{w|G|c}$  and  $\boldsymbol{\rho}_{w|G|c}$  denote the operational space inertia, the Coriolis and centrifugal force, and gravity force for the wheel motion task (see Appendix B). The feed-forward torque can be calculated by the operational space forces of the designed tasks as follows,

$$\boldsymbol{\tau} = \mathbf{P}\left[\mathbf{J}_c^T\mathbf{F}_c + \mathbf{J}_{G|c}^T\mathbf{F}_{G|c} + \mathbf{J}_{w|G|c}^T\mathbf{F}_{w|G|c}\right] = \mathbf{P}\left[\mathbf{J}_{G|c}^T\mathbf{F}_{G|c} + \mathbf{J}_{w|G|c}^T\mathbf{F}_{w|G|c}\right]. \quad (35)$$

Then we combine the feed-forward and feedback torques in (35) and (32) to achieve the actuated torque  $\boldsymbol{\tau}_{act}$  as,

$$\boldsymbol{\tau}_{act} = \boldsymbol{\tau} + \boldsymbol{\tau}_{fb}, \quad (36)$$

where  $\boldsymbol{\tau}_{act}$  is applied on the robot actuators (cf. Fig. 4) to enable the robot to track the centroidal motion and the wheel motion references compliantly on simulated various unknown rough terrains. In the operational space control,



the contact constraint has the highest priority. The contact force  $F_c$  can be estimated by multiplying  $\bar{J}_c^T$  at both sides of the generalized dynamics model in (3) and by combining the contact constraints in (4), as below,

$$F_c = -\bar{J}_c^T S^T \tau_{act} + \mu_c + \rho_c, \quad (37)$$

where  $\mu_c$  and  $\rho_c$  denote the operational-space Coriolis and centrifugal force, and the gravity force at the wheel-contact points (see Appendix B). By using this method, the force sensors are not necessary. Although the contact forces are not accurate when the contact-point positions are not estimated precisely, the prioritized impedance controller can compensate the influence to the dynamics model. The estimation accuracy of the contact force can be improved by one more accurate estimation method of the contact-point positions.

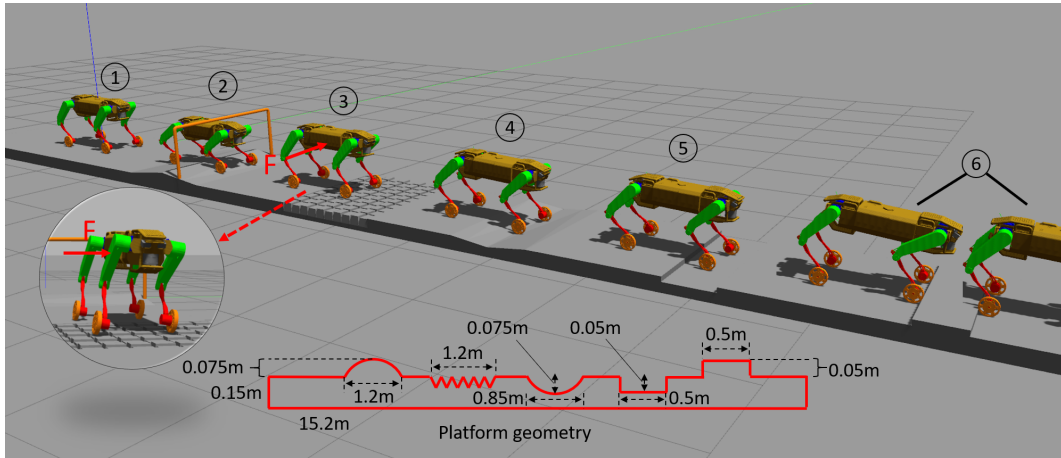
In this section, the feed-forward torque controller and the feedback prioritized impedance controller always satisfy the contact constraints. When the robot runs on the rough terrain and the centroidal acceleration reference in the vertical direction does not change with big amplitude, the contact unilateral limit is satisfied. In addition, when the rolling friction is enough, the friction cone is also respected. Therefore, the fulfillment of the contact constraints by the torque controller ensures the robot stability to resist external disturbances, including the unknown rough terrains, the external force, etc, shown in the complex simulated scenarios in Section 5. Furthermore, we also developed one optimization based torque controller in [32] which can handle the contact situation with inequality constraints, e.g. the contact unilateral limit, the friction cone and the actuated-joint limits. In [32], we have verified the optimization based torque controller using our wheel motion generator in this paper.

## 5. Simulation Results on Various Rough Terrains

The motion generator proposed in Section 3 is tested in two simulation scenarios (see Fig. 5 and Fig. 7) using our virtual quadruped-on-wheel robot, “TowrISIR” in Fig. 1. The simulation results are reported using the physical engine established by ROS-GAZEBO, in which we use several virtual sensors, including GPS for the real-time robot position, IMU for the base orientation and angular velocities, joint position encoders and torque sensors for joint state measurements. Based on the sensor-feedback data, we calculate the base and the centroidal properties. These robot states are applied in the impedance controllers and the motion generators, shown in Fig. 4. We set the centroidal motion reference in the sagittal plane and its lateral reference is always zero. The simulations on various rough terrains are used to verify the validity, adaptiveness and usefulness of our proposed adaptive wheel motion generator. The motion simulation is based on the general control framework presented in Fig. 4 that combines the feedback controller in the impedance input-port and the feed-forward controller that computes the required torque vector which satisfies the Equation of Motion, the contact constraints and the task hierarchy. In each control loop, the wheel motion generator outputs the desired wheel-joint position, velocity and acceleration, and the centroidal angular velocity and acceleration are also achieved, detailed in Section 2 and 3. The desired whole-body configuration and velocities are used in the feedback prioritized impedance controller in Section 4.1 and 4.2. Then the desired centroidal and wheel-joint accelerations serve as the inputs of the hierarchical torque controller in Section 4.3. The calculated torque respects the contact constraints and it is applied on our quadruped-on-wheel robot to run on various rough terrains in this section.

### 5.1. Running on Artificial Environment

In the first scenario in Fig. 5, only the door-like horizontal bar position is known and the robot crosses over several kinds of artificial terrains, including one bump, one hollow and two stairs. To validate the effectiveness and adaptivity of our wheel motion generator, one bump and the horizontal bar are put at the same place in stage 2, and the desired CoM height is controlled to enable the robot to squat down and rolls over the bump and through the bar. As described in Section 4.1, we set high stiffness and damping on the height control of the centroidal translational motion, and the components of the stiffness and damping for the wheel-center motion in the base vertical direction are set with small values. In stage 3, one virtual lateral external force  $F = 180N$  is applied on the robot which begins at 9s and lasts 0.2 seconds. Since the rough environment is unknown and the contact points are always estimated at the lowest points of the wheels in Section 2.2, we can see that the wheel tracking in Fig. 6(c) experiences a relatively big fluctuation. It is not very precise when the robot runs on the continuous bump and hollow (in stage 2 and 4), stones (in stage 3) and discontinuous stairs (in stage 5 and 6). The inaccurate contact-point positions induce inaccuracy to the whole-body kinematics model in Section 2.2 and the generalized dynamics model in Section 2.1. Then the imprecise models influence the outputs of the motion generator in Section 3 and the results of the centroidal forward motion tracking, especially in stage 5 and 6 with discontinuous stairs. Even though, the desired and actual CoM positions are still



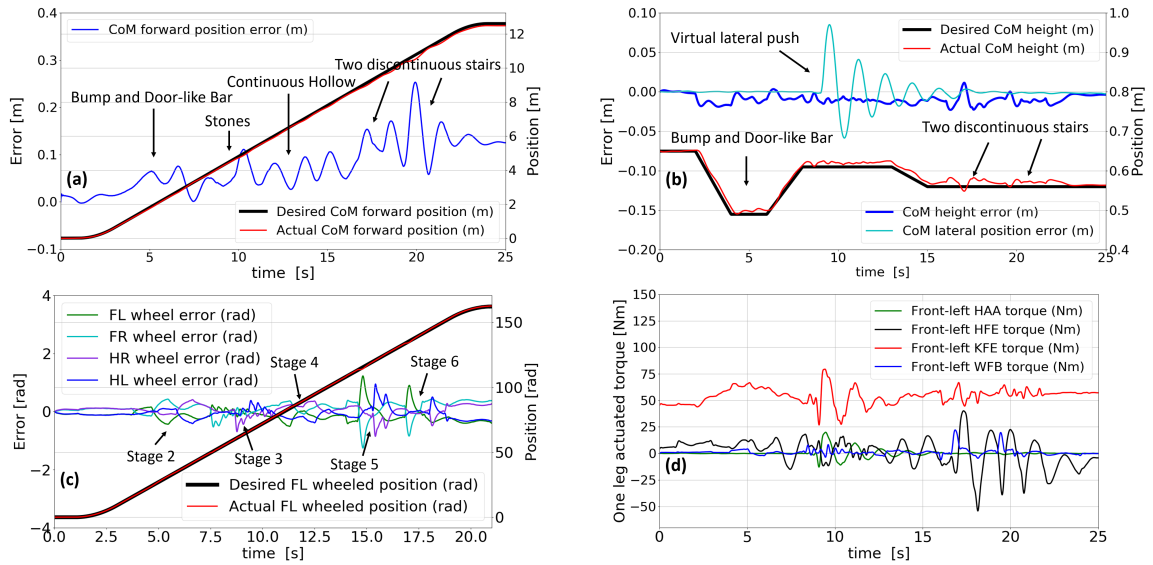
**Figure 5:** The robot runs forward on various rough terrains and handles several activities. The platform  $height = 0.15m$ ,  $length = 15.2m$ , and the rough terrain geometry is depicted. The whole running process is divided into six stages. [Stage 1]: the beginning state. [Stage 2]: the robot squats down to cross over one bump and through one door-like horizontal bar. [Stage 3]: the robot is driven on stones with different heights (the max difference is  $0.02m$ ) and the robot is pushed by one lateral external force  $F$  in the zoom-in picture. [Stage 4]: the robot crosses one continuous hollow. [Stage 5 and 6]: the robot climbs over two kinds of stairs.

almost coincident in the long running process, shown in Fig. 6(a). For the squatting motion in stage 2 and the push-recovery motion in stage 3, the adaptive wheel motion is generated to track the CoM motion according to the real-time leg states, the varied CoM height and its lateral position (see Fig. 6(b)), because the actual legged motion is used to achieve the wheel velocity reference, and the actual legged configuration/velocity is applied to derive the wheel acceleration reference, and these strategies enable the robot to adapt to the rough terrain with robustness, detailed in Section 3. By using the whole-body torque control (in Section 4.3) integrated with our prioritized impedance controller in Section 4.2, the robot is enabled to run on various rough terrains and track the centroidal motion with compliance, especially for the periods with the stairs in stage 5 and 6, and after the push in stage 3, the robot can recover to the normal state, shown in Fig. 6(b). In addition, we plot the actuated torque for the joints of the front-left leg in Fig. 6(d), which is generated by our compliant torque controller in Section 4. It is noticed that the HAA torque is always very small except during the push-recovery motion. The wheel joint torque experience a small fluctuation when the robot runs on the stones, the value becomes bigger when the robot crosses over the two discontinuous stairs. In the future work, we will develop one estimation method for the wheel-contact positions to improve the tracking performance. In addition, we developed one optimization based torque controller in [32] to achieve several operational tasks which are achieved by respecting the contact constraints. The inequality constraints include the base orientation and position limits, the actuated-joint position and velocity limits, the unilateral contact constraint and the friction cone. The torque controller in [32] is verified on our robot with one manipulator on its base using our wheel motion generator in this paper. In these works, the contact constraints always have the highest priority which ensure the robot stability while handling several activities.

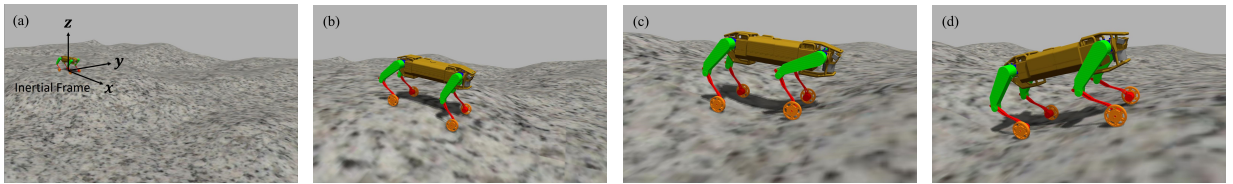
## 5.2. Running on 3D Rough Terrain

The second scenario relates to the validation of the robot running on numerous slopes in the unknown continuous 3D rough environment, shown in Fig. 7(a). On this artificial terrain with complex slopes, the stiffness and damping for the centroidal height control is set to be low and we give more concern on the wheel-center motion relative to the base. The robot runs downward and upward to track the whole-body centroidal motion reference. The prioritized impedance controller enables the robot to do active legged suspension when the robot starts to go downward in Fig. 7(b) or prepares the climb-upward motion in Fig. 7(c). Finally, the robot climbs over the slope in Fig. 7(d). The CoM and wheel positions and errors are plotted in Fig. 8(a) and Fig. 8(b) respectively. Even with different legged suspension and configuration, our method in Section 3 can still generate the adaptive wheel motion to track the centroidal motion. Along the whole process on the complex rough terrain, the tracking performance is quite good with small errors even though the contact point is not estimated precisely. With the prioritized impedance controller based whole-body torque

## Rolling based Locomotion on Rough Terrain for a Wheeled Quadruped using Centroidal Dynamics



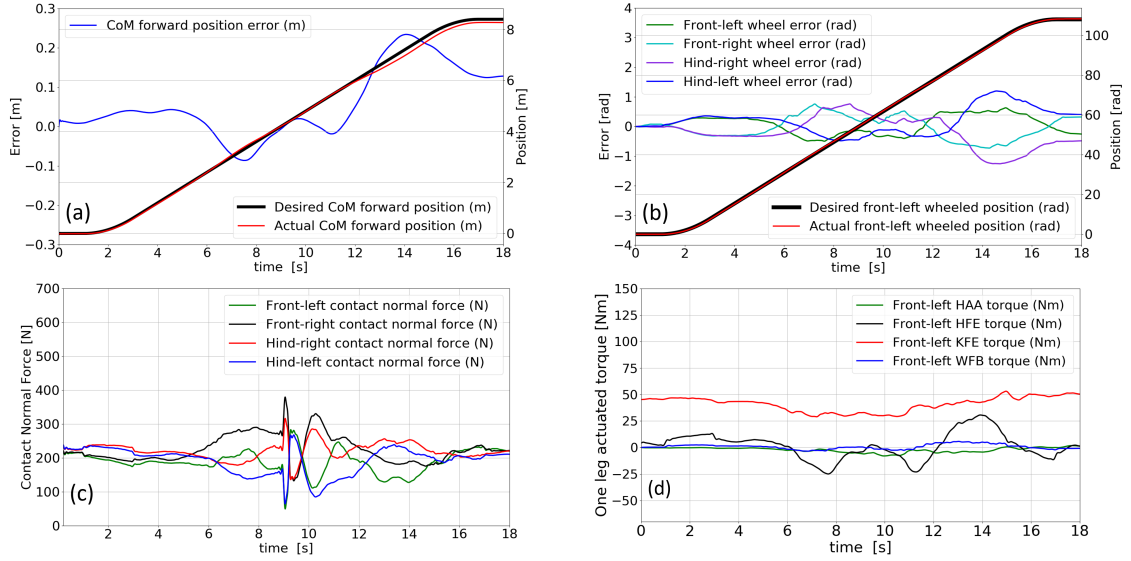
**Figure 6:** Motion tracking results of the centroidal motion and the wheel motion. The left and right axes represent the errors and positions respectively. (a) records the robot centroidal forward position and error. (b) plots the lateral position error and its vertical height/error. (c) shows the four-wheel rotational positions and errors. Since the four-wheel motion experience the almost coincident trajectories, only the desired and actual front-left wheel positions are plotted. (d) plots the actuated-joint torque for the front-left leg.



**Figure 7:** The robot is driven on unknown 3D rough terrain with various slopes. (a) is the beginning state. In (b), the robot runs downward one slope. In (c) and (d), the robot prepares and climbs upward respectively. (b) and (c) show the legged suspension with compliance.

control in Section 4, the required actuated torque is generated to track the motion references. The contact constraint with the highest priority is ensured, shown in Fig. 8(c) which shows that the four legs are always in contact with the ground. The four normal contact forces are around  $200N$  using the estimation method in Section 4.3. In Fig. 8(d), we plot the actuated torque for the joints of the front-left leg, which is generated by our compliant torque controller in Section 4. It is noticed that compared with the generated joint torque in Section 5.1, all joint torque is smooth along the whole process when the robot runs on this 3D continuous terrain. Only the HFE joint torque experiences a small fluctuation when the robot runs downward one slope at  $6.3s$  and climbs upward one slope at  $12s$ .

To sum up, the simulations verify that the proposed wheel motion generator (combining the kinematics model and the centroidal momentum/dynamics model) is adaptive and effective for our quadruped-on-wheel robot when running on various unknown rough terrains. The prioritized impedance controller based multi-task operational space controller enables the robot with compliance abilities, especially for active legged suspension and the push-recovery motion. The two simulations are done on a usual PC with a  $i7-7700$  CPU,  $3.6GHz$ . The sampling time is fixed with  $1ms$ . The calculation period for the motion generator and the torque controller is max  $0.7ms$ , about  $1429Hz$  which can be suitable for real-time control on other legged-wheeled robots with high DOFs. The simulation results provided by our approach are quite satisfactory, which can also be seen in the attachment video. With the proposed motion generator and the compliant torque controller, our robot can be used to work for agriculture, industries and mountain lanes. One or two manipulators can be assembled on the floating base to achieve more complex grasping tasks.



**Figure 8:** Simulation results of the second scenario. (a) records the quadruped CoM motion in forward direction. (b) plots the four wheels' rotational position errors along the whole process. Only the desired and actual front-left wheel positions are plotted. (c) shows the four wheel contact normal forces. (d) plots the actuated-joint torque for the front-left leg.

## 6. Conclusions & Future Work

In this paper, we propose one wheel motion generator to track the centroidal motion of one quadruped-on-wheel robot, by combining the whole-body inverse kinematics model and the centroidal momentum/dynamics model. The inverse kinematics model is derived by using the wheel contact constraints and the wheel motion is extracted out depending on the base motion and the legged motion. To track the robot centroidal motion reference, the whole-body inverse kinematics model and the centroidal momentum/dynamics model are mathematically decomposed into several components relating to the base, the legged and the wheel motions, in which several algorithms are developed. Then the two models are combined to achieve the adaptive wheel positions, velocities and accelerations. By using the prioritized torque control framework mixed with our proposed prioritized impedance controller, the contact constraints are ensured and our robot is successfully driven to track the centroidal motion reference on various simulated rough terrains, which validate the adaptivity and effectiveness of our wheel motion generator.

In this work, all the wheels are always in contact with the ground. We set the desired legged-joint velocity and acceleration to be zero and they serve as the inputs of the wheel motion generator. In the future works, we will develop one general strategy to generate online the adaptive legged motion and the wheel motion together, especially for the fast squatting-down motion. Then we will combine the dynamic legged and wheel motions together to achieve more complex locomotion modes, e.g. trotting-rolling, jumping-rolling etc.

## APPENDIX A

Since we use spatial vectors in the system modeling and the wheel motion generators, the translational base and centroidal motions expressed in their own frames can be achieved as below,

$$\mathbf{v}_b = {}^b \mathbf{R}_0 \dot{\mathbf{p}}_b, \quad (38a)$$

$$\dot{\mathbf{v}}_b = {}^b \mathbf{R}_0 \ddot{\mathbf{p}}_b - \boldsymbol{\omega}_b \times \mathbf{v}_b, \quad (38b)$$

$$\mathbf{v}_G = \dot{\mathbf{p}}_G, \quad (38c)$$

$$\dot{\mathbf{v}}_G = \ddot{\mathbf{p}}_G - \boldsymbol{\omega}_G \times \mathbf{v}_G, \quad (38d)$$

where  ${}^b \mathbf{R}_0$  transform 3D vectors from the inertial frame to the base frame, which is equal to  ${}^b \mathbf{R}_G$ .

## APPENDIX B

In the contact-consistent prioritized multi-task torque control framework [17], the contact constraint and force are already integrated in the derivation for the operational space control of the other designed tasks. In this paper, the contact-consistent dynamically inversed Jacobians of the three designed tasks are listed as below,

$$\bar{\mathbf{J}}_c = \mathbf{M}^{-1} \mathbf{J}_c^T (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T)^{-1}, \quad (39a)$$

$$\bar{\mathbf{J}}_{G|c} = \mathbf{M}^{-1} \mathbf{J}_{G|c}^T (\mathbf{J}_{G|c} \mathbf{M}^{-1} \mathbf{J}_{G|c}^T)^{-1}, \quad (39b)$$

$$\bar{\mathbf{J}}_{w|G|c} = \mathbf{M}^{-1} \mathbf{J}_{w|G|c}^T (\mathbf{J}_{w|G|c} \mathbf{M}^{-1} \mathbf{J}_{w|G|c}^T)^{-1}. \quad (39c)$$

The functions relating to the operational space forces of the two tasks in (33) and (34) are derived by multiplying  $\bar{\mathbf{J}}_{G|c}^T$  and  $\bar{\mathbf{J}}_{w|G|c}^T$  respectively on both sides of the generalized dynamics model in (3). The variables in (33), (34) and (37) are listed in (40), (41) and (42) respectively as follows.

$$\Lambda_{G|c} = (\mathbf{J}_{G|c} \mathbf{M}^{-1} \mathbf{J}_{G|c}^T)^{-1}, \quad (40a)$$

$$\boldsymbol{\mu}_{G|c} = \Lambda_{G|c} \mathbf{J}_G \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{C} + \Lambda_{G|c} \mathbf{J}_G \bar{\mathbf{J}}_c \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \Lambda_{G|c} \dot{\mathbf{J}}_G \dot{\mathbf{q}}, \quad (40b)$$

$$\boldsymbol{\rho}_{G|c} = \Lambda_{G|c} \mathbf{J}_G \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{G}. \quad (40c)$$

$$\Lambda_{w|G|c} = (\mathbf{J}_{w|G|c} \mathbf{M}^{-1} \mathbf{J}_{w|G|c}^T)^{-1}, \quad (41a)$$

$$\boldsymbol{\mu}_{w|G|c} = \Lambda_{w|G|c} \mathbf{J}_w \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{C} + \Lambda_{w|G|c} \mathbf{J}_w \bar{\mathbf{J}}_c \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \Lambda_{w|G|c} \dot{\mathbf{J}}_w \dot{\mathbf{q}}, \quad (41b)$$

$$\boldsymbol{\rho}_{w|G|c} = \Lambda_{w|G|c} \mathbf{J}_w \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{G}. \quad (41c)$$

$$\Lambda_c = (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T)^{-1}, \quad (42a)$$

$$\boldsymbol{\mu}_c = \bar{\mathbf{J}}_c^T \mathbf{C} - \Lambda_c \dot{\mathbf{J}}_c \dot{\mathbf{q}}, \quad (42b)$$

$$\boldsymbol{\rho}_c = \bar{\mathbf{J}}_c^T \mathbf{G}. \quad (42c)$$

where  $\Lambda_c$  denotes the operational space inertia at the wheel-contact points.

## APPENDIX C

For one general null-space projector  $\mathbf{N} \in \mathbb{R}^{n \times n}$ , it has the following properties,

$$\mathbf{N} \mathbf{N} = \mathbf{N}, \quad (43a)$$

$$\mathbf{N} \mathbf{M}^{-1} = \mathbf{M}^{-1} \mathbf{N}^T = \mathbf{N} \mathbf{M}^{-1} \mathbf{N}^T. \quad (43b)$$

In addition, for the designed three tasks in this paper, the null-space projectors have the following properties,

$$\mathbf{N}_{G|c} = \mathbf{N}_{G|c} \mathbf{N}_c, \quad (44a)$$

$$\mathbf{N}_{all} = \mathbf{N}_{all} \mathbf{N}_{G|c} = \mathbf{N}_{all} \mathbf{N}_{G|c} \mathbf{N}_c, \quad (44b)$$

$$\mathbf{J}_c \mathbf{N}_c = \mathbf{0}_{3n_c \times n}, \quad \mathbf{J}_{G|c} \mathbf{N}_{G|c} = \mathbf{0}_{6 \times n}, \quad \text{and} \quad \mathbf{J}_{w|G|c} \mathbf{N}_{all} = \mathbf{0}_{n_c \times n}. \quad (44c)$$

Therefore, for the impedance forces of the designed tasks in the operational space in (31), we give the derivations for two examples as follows,

$$\bar{\mathbf{J}}_c^T \mathbf{J}_{w|G|c}^T = (\Lambda_c \mathbf{J}_c \mathbf{M}^{-1}) (\mathbf{J}_w \mathbf{N}_{G|c})^T = (\Lambda_c \mathbf{J}_c \mathbf{M}^{-1}) (\mathbf{J}_w \mathbf{N}_{G|c} \mathbf{N}_c)^T = (\Lambda_c \mathbf{J}_c \mathbf{N}_c \mathbf{M}^{-1}) (\mathbf{J}_w \mathbf{N}_{G|c})^T = \mathbf{0}_{3n_c \times n_c}, \quad (45a)$$

$$\bar{\mathbf{J}}_{w|G|c}^T \mathbf{J}_{G|c}^T = (\Lambda_{w|G|c} \mathbf{J}_w \mathbf{M}^{-1}) \mathbf{J}_{G|c}^T = (\Lambda_{w|G|c} \mathbf{J}_w \mathbf{N}_{G|c} \mathbf{M}^{-1}) \mathbf{J}_{G|c}^T = (\Lambda_{w|G|c} \mathbf{J}_w \mathbf{M}^{-1}) \mathbf{N}_{G|c}^T \mathbf{J}_{G|c}^T = \mathbf{0}_{n_c \times 6}. \quad (45b)$$

## References

- [1] Popovic M B, Goswami A, Herr H. Ground reference points in legged locomotion: Definitions, biological trajectories and control implications[J]. *The international journal of robotics research*, 2005, 24(12): 1013-1032.
- [2] Orin D E, Goswami A. Centroidal momentum matrix of a humanoid robot: Structure and properties. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008: 653-659.
- [3] 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.
- [4] Lee S H, Goswami A. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground[J]. *Autonomous Robots*, 2012, 33(4): 399-414.
- [5] Pucci D, Nava G, Nori F. Automatic gain tuning of a momentum based balancing controller for humanoid robots. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). IEEE, 2016: 158-164.
- [6] Ugurlu B, Havoutis I, Semini C, et al. Pattern generation and compliant feedback control for quadrupedal dynamic trot-walking locomotion: experiments on RoboCat-1 and HyQ. *Autonomous Robots*, 2015, 38(4): 415-437.
- [7] Grand Christophe, Benamar Faiz, 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.
- [8] Bouton A, Grand C, Benamar F. Design and Control of a Compliant Wheel-on-Leg Rover which Conforms to Uneven Terrain[J]. *IEEE/ASME Transactions on Mechatronics*, (2020), doi: <http://doi.org/10.1109/TMECH.2020.2973752>
- [9] 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.
- [10] Klamt, Tobias, and Sven Behnke. Anytime hybrid driving-stepping locomotion planning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017: 4444-4451.
- [11] An S, Oh Y, Kwon D S. Zero-moment point based balance control of leg-wheel hybrid structures with inequality constraints of kinodynamic behavior. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 2471-2477.
- [12] Suzumura A, Fujimoto Y. Real-time motion generation and control systems for high wheel-legged robot mobility[J]. *IEEE Transactions on Industrial Electronics*, 2013, 61(7): 3648-3659.
- [13] Nagano K, Fujimoto Y. The stable wheeled locomotion in low speed region for a wheel-legged mobile robot[C]//2015 IEEE International Conference on Mechatronics (ICM). IEEE, 2015: 404-409.
- [14] 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.
- [15] Kamedula M, Kashiri N, Tsagarakis N G. Wheeled motion kinematics and control of a hybrid mobility CENTAURO robot[J]. *Robotics and Autonomous Systems*, (2020), doi: <https://doi.org/10.1016/j.robot.2020.103482>.
- [16] 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.
- [17] Park J, Khatib O. Contact consistent control framework for humanoid robots. *Robotics and Automation*, 2006. *ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006: 1963-1969.
- [18] Sentis L, Park J, Khatib O. Compliant control of multicontact and center-of-mass behaviors in humanoid robots[J]. *IEEE Transactions on robotics*, 2010, 26(3): 483-501.
- [19] 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.
- [20] Orsolino R, Focchi M, Mastalli C, et al. Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots[J]. *IEEE Robotics and Automation Letters*, 2018, 3(4): 3363-3370.
- [21] Caron S, Pham Q C, Nakamura Y. Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics. *Robotics: science and systems*. 2015, 11: 1-9.
- [22] Dietrich A, Wimbock T, Albu-Schaffer A, et al. Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom[J]. *IEEE Robotics & Automation Magazine*, 2012, 19(2): 20-33.
- [23] David E Orin, Ambarish Goswami and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, vol. 35, no. 2-3, pages 161-176, 2013.
- [24] Garofalo G, Henze B, Engelsberger J, et al. On the inertially decoupled structure of the floating base robot dynamics. *IFAC-PapersOnLine*, 2015, 48(1): 322-327.
- [25] Nava G, Romano F, Nori F, et al. Stability analysis and design of momentum-based controllers for humanoid robots. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 680-687.
- [26] Mastalli C, Havoutis I, Focchi M, et al. Hierarchical planning of dynamic movements without scheduled contact sequences. 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 4636-4641.
- [27] Dafarra S, Romano F, Nori F. Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub humanoid robot. 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids). IEEE, 2016: 152-157.
- [28] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.
- [29] Miomir Vukobratovic and Branislav Borovac. Zero-moment point—thirty five years of its life. *International journal of humanoid robotics*, vol. 1, no. 01, pages 157-173, 2004.
- [30] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, vol. 5, no. 1, pages 90-98, 1986.
- [31] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.
- [32] Du W, Benamar F. A compact form dynamics controller for a high-DOF tetrapod-on-wheel robot with one manipulator via null space based convex optimization and compatible impedance controllers[J]. *Multibody System Dynamics*, 2020: 1-17.

- [33] Mistry M, Buchli J, Schaal S. Inverse dynamics control of floating base systems using orthogonal decomposition. 2010 IEEE international conference on robotics and automation. IEEE, 2010: 3406-3412.
- [34] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, D. G. Caldwell, Design of the Hydraulically-Actuated, Torque-Controlled Quadruped Robot HyQ2Max, IEEE/ASME Transactions on Mechatronics, 1-1, 2016
- [35] Du, W., Fnadi, M., Benamar, F. Integration of Prioritized Impedance Controller in Hierarchical Operational-Space Torque Control Frameworks for Locomotion Robots. Robotics and Autonomous Systems (2020). Under review.