



HAL
open science

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

Wenqian Du, Faïz Ben Amar

► **To cite this version:**

Wenqian Du, Faïz Ben Amar. 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. *Multibody System Dynamics*, 2020, 49 (4), pp.447-463. 10.1007/s11044-020-09728-y . hal-03177955

HAL Id: hal-03177955

<https://hal.science/hal-03177955>

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.

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

Wenqian DU · Faïz Benamar

Received: date / Accepted: date

Abstract This paper develops a compact form dynamics controller to generate multi-compliant behaviors for a new designed Tetrapod-on-wheel robot with one manipulator. The whole-body compliant torque controller is stated through one null space based convex optimization and compatible null space based impedance controllers. Different with fixed contact points of conventional quadruped robots, the kinematic wheel contact constraints are derived for our legged-on-wheel robot, which serves as the basis for each task reference extraction and each compliance controller. The compact relationships between task references and optimization control variables are extracted using null space based inverse dynamics, which is used to build the cost function in the operational space and/or in the joint space. The whole-body control frame is developed and several null space based feed-back impedance controllers are integrated into the compact relationships to allow the robot to achieve compliance and compensate the model impreciseness, especially the wheel contact model. Then the detailed algorithm is presented whose output combines the feed-forward and feedback torque. The validation of our approach is performed via advanced numerical simulations for a virtual legged-on-wheel robot with one manipulator.

Keywords Compact-form dynamics controller · Tetrapod-on-wheel robot · Null space · Convex Optimization · Compatible Impedance Control · Wheel Contact Model

1 Introduction

Conventional wheel robots have great performance and potential on continuous terrain with high speed and less energy consuming, but easily stuck on rough terrain with steep obstacles.

F. Wenqian DU
Institut des Systèmes Intelligents et de Robotique: ISIR, Faculty of Science & Engineering, Sorbonne University, 75005 Paris, France
Tel.: +33-699-28-00-48
E-mail: wenqian.du@isir.upmc.fr

S. Faïz Benamar
Institut des Systèmes Intelligents et de Robotique: ISIR, Faculty of Science & Engineering, Sorbonne University, 75005 Paris, France
E-mail: amar@isir.upmc.fr

In contrast, legged robots are not tied to discontinuous natural environments like mountain lanes, or artificial paths like stairs and can move with agility to cope with uncertainties or disturbances. To allow mobile robots with more abilities and energy efficiency, legged-on-wheel robots combine strengths of autonomous vehicles and legged robots, which can be used for universe exploration, industry logistics, disaster rescue and as assistants in daily life. However, it faces more complexity on adaptive control for locomotion considering high degrees of freedoms (DOF) and various constraints, and for compatibility between wheel motion and legged motion. Compared with inverse kinematics control of operational space tasks like for locomotion and posture, dynamics methods allow much more elaborated control concepts and get far more natural abundant behaviors. One of the widely used model based inverse dynamics is developed by [1] in operational space, which is used to compute the generalized forces necessary to achieve desired trajectories for end-effectors. Authors in [2] propose one more specific control frame that can synthesize whole-body behaviors simultaneously which extends the operational space controllers for robots with floating bases. However, this method cannot integrate inequality constraints, like friction, joint limits and torque limits.

Another approach to solve the inverse dynamics is to use optimization techniques which can deal with more general constraints with equality and/or inequality formulations. The most popular approaches are with the quadratic programming (QP) which can handle multi tasks and constraints at the same time, including the weighted QP and the hierarchical QP. Weight-based schemes combine tasks in an integrated QP scheme by distributing weights \mathbf{w} to each of them. The general objective form for n_i tasks becomes $\sum_{i=1}^{n_i} \|\ddot{e}_i^* - \ddot{e}_i\|_{\mathbf{w}}^2$ [3] and [4], in which \ddot{e}_i^* and \ddot{e}_i represent respectively the task references and their expectations. However, when the redundancy is not enough, weighted tasks in this scheme may conflict with each other, resulting that no tasks can be fully satisfied but obtained with a compromise [5]. Furthermore, [8] proposes the Hierarchical Quadratic Program (HQP) solver that extends the traditional cascade of QPs so that equality and inequality constraints can be handled at any level of the cascade. HQP is based on prioritized schemes which divide the multiple tasks into hierarchies ensuring lower priority tasks cannot interfere with higher priority ones. In this way, the top priority task is solved initially and the task error serves as the constraint for the next priority task, until the lowest one, which treats the problem recursively [6]. Moreover, [7] combines weighting-based and prioritized schemes together to treat optimization of tasks into hierarchies and assigns weights to tasks in each level of hierarchy. In addition, [5] generates different types of motions (such as sitting, climbing, yoga motion, walking on rough terrain) by applying HQP and Stack of Tasks (SoT) that allows to add or remove tasks at any instant of time. A prioritization scheme based on a conic optimization to control the centroidal angular momentum for generating movements like kick and jump has been introduced in [9] and [10]. Authors in [11] and [12] apply HQP for a quadrupedal robot to achieve centroidal trajectory optimization and non-slipping by contact force optimization. Recently, one null space based hierarchical QP is proposed in [18] in which fewer equality constraints are achieved and are added to the cascade of QPs. Although hierarchical QP framework can ensure strict priority between each task level and generates required torque for complex behaviors subjected to different kinds of constraints, the hierarchies are not easy to distinguish. What's more, when the robot has high DOF, the slow calculation loop limits robots for highly dynamics motion. Therefore, weighted optimization framework is more suitable by tuning importance weights for multi-dynamics behaviors.

For conventional weighted convex optimization without null space projections, when there is not enough DOF for the robot to finish all tasks feasibly, unfeasible motion directions cannot be removed from the tasks. Furthermore, conventional feedback based controllers can

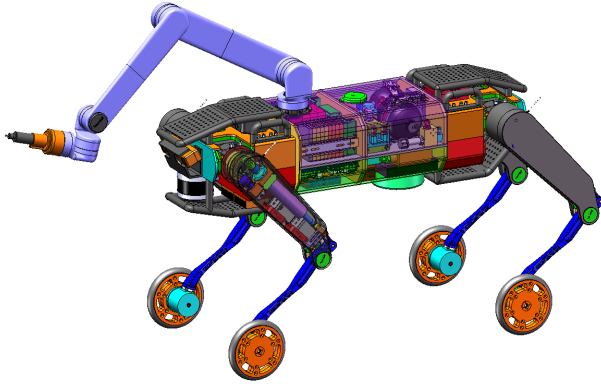


Fig. 1 TowrISIR—A tetrapod-on-wheel robot designed in ISIR lab has the similar structure with the quadruped robot HyQ2Max without wheels [19]. TowrISIR is with length $l^r \approx 1.06m$, width $w^r \approx 0.55m$ and height $h^r \approx 0.8m$, equipped with 12 hydraulic actuators and 4 wheel motors. Each leg has 4 joints, including hip abduction/adduction joint (HAA), hip flexion/extension joint (HFE), knee flexion/extension joint (KFE) and wheel forward/backward joint (WFB). Each joint is integrated with force/torque sensor and absolute/incremental encoder.

influence all tasks together which cannot be accepted. These two problems can be avoided in this paper by using null space projections, which is always be applied in hierarchical optimization framework, but neglected in the weighted theme.

In this paper, we build one compact form dynamics controller for a tetrapod-on-wheel robot using one null space based convex optimization. The null space based methods are applied for task reference extractions, the cost function and constraint establishment, and compliance controllers. In our null space based approach, decomposition methods can be applied to select feasible motion direction for specific tasks, and impedance controllers can be consistent with task priorities. We focus on how to use a null space based weight optimization and several compatible null space based impedance controllers for a dedicated tetrapod-on-wheel robot with one manipulator. The relationships between task references and the joint actuated torque are extracted using null space based inverse dynamics, which is applied to build the cost function and express various constraints using the actuated torque. To allow contact forces integrated in the cost function and various constraints, the wheel contact model is derived using spatial vectors, different with fixed contact points for quadrupeds. Finally several compatible null space based impedance controllers are developed and integrated in the convex optimization to track task references and compensate model impreciseness, especially the wheel contact model.

The reminder of this paper is organized: In section 2, inverse dynamics in operational space is firstly stated to extract the compact relationship between each task and the required torque in joint space. Then the null space based convex optimization in operational space and joint space are detailed in Section 3. Then the whole-body control framework and null space based compliance controllers are developed. In section 4, we validate and discuss our method through numerical simulation under ROS-GAZEBO with our virtual legged-on-wheel robot.

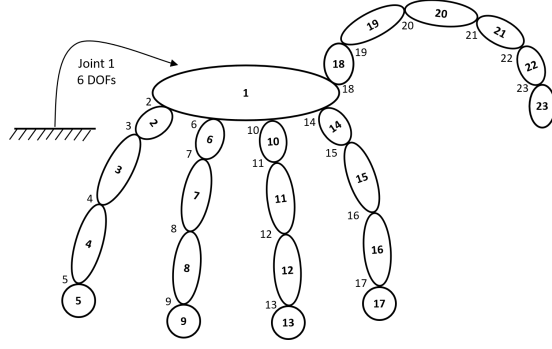


Fig. 2 Connectivity of TowrISIR: number inside each circle represents body index. and outside numbers mean joint index. Joint 1 represents the floating-base joint which has 6 DOFs. There are $N_B = 23$ bodies in which bodies from 18 to 23 represent the manipulator and bodies from 1 to 17 represent the quadruped.

2 System Modeling

TowrISIR—A new tetrapod-on-wheel robot with one manipulator is designed in ISIR laboratory (see in Fig. 1). It is equipped with one floating base and four branches, each branch has three leg joints and one wheel joint. From a practical point of view, tetrapod-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 motion & torque control. The more wheel joints induce more complexity of the kinematics/dynamics model and the contact constraints, which will be detailed in this section. Spatial vectors [9] are used through this paper written in upright bold letters, and conventional 3D vectors are italic. .

2.1 Dynamics Model

This robot has $N_B = 23$ bodies and has $n = 28$ DOF in which 6 DOF is relative to the spatial floating base motion (see Fig. 2). The robot configuration is represented as

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

where \mathbf{x}_b and \mathbf{q}_r define the gait configuration of the quadrupedal motion, $\mathbf{x}_b = [\boldsymbol{\theta}_b^T, \mathbf{p}_b^T]^T$ is the orientation and position of the floating base in inertial global coordinate, $\mathbf{q}_r \in \mathbb{R}^{n_r}$ represents n_r revolute joint configurations, $\mathbf{q}_w \in \mathbb{R}^{n_w}$ represents n_w continuous wheel joint configurations, $\mathbf{q}_m \in \mathbb{R}^{n_m}$ represents n_m revolute arm joint configurations.

The generalized dynamics model in joint space is derived by Newton-Euler method using connectivity shown in Fig. 2. The equations are achieved by spatial vectors [16] with bold-straight letters as follows,

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

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

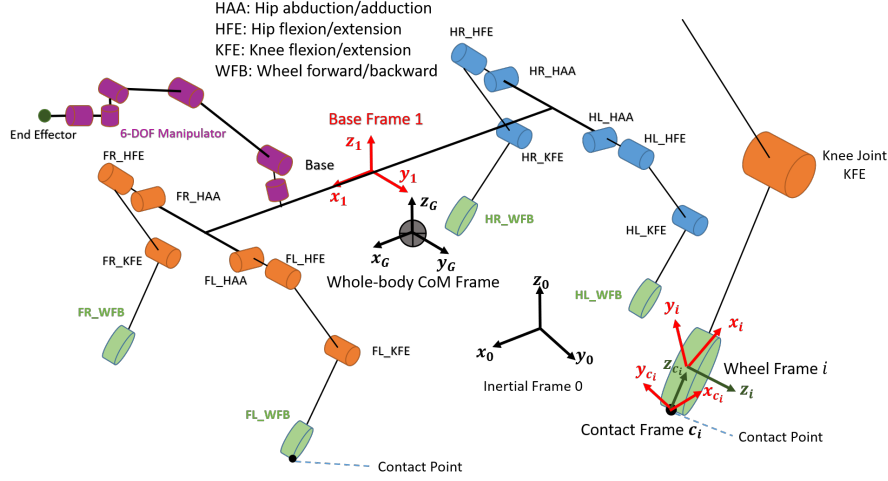


Fig. 3 Frame Definition: The quadruped CoM frame G is set parallel with the base frame 1. The frames relating to the inverse kinematics are zoomed in, including the contact frame c_i , wheel frame i and the knee frame $p(i)$. F, L, R, H represent front, left, right and hind.

In addition, in this paper, we will control the CoM motion of the quadruped parts which include $N_q = 17$ bodies from index 1 to 17, as noticed in Fig. 2. This detailed structure of our robot is shown in Fig. 3. The Jacobian at the quadruped CoM is derived as

$$\mathbf{J}_G = \mathbf{I}_G^{-1} ({}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M}_{quad}) = [{}^1\mathbf{X}_G^T (\mathbf{U} \mathbf{M}_{quad} \mathbf{U}^T) {}^1\mathbf{X}_G]^{-1} ({}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M}_{quad}), \quad (3)$$

$$\mathbf{M}_{quad} = \sum_{i=1}^{N_q} \mathbf{J}_i^T \mathbf{I}_i \mathbf{J}_i, \quad (4)$$

where $\mathbf{I}_G \in \mathbb{R}^{6 \times 6}$ represents the composite rigid body inertia at the quadruped CoM. $\mathbf{U} = [\mathbf{1}_{6 \times 6}, \mathbf{0}_{6 \times (n-6)}]$ is a selection matrix. $\mathbf{J}_i \in \mathbb{R}^{6 \times n}$ and $\mathbf{I}_i \in \mathbb{R}^{6 \times 6}$ are the Jacobian and inertia respectively at link i 's frame origin. It is noticed that $\mathbf{M}_{quad} \in \mathbb{R}^{n \times n}$ is the quadruped mass matrix which is different with the whole-body mass matrix \mathbf{M} . \mathbf{M}_{quad} is derived from the inertias of bodies from 1 to 17 which means that components in \mathbf{M}_{quad} relating to the manipulator bodies are equal to zero. \mathbf{M}_{quad} is a by-product when getting whole-body generalized mass matrix \mathbf{M} through the Composite Rigid Body algorithm [22]. ${}^1\mathbf{X}_G$ is the transformation matrix from quadruped CoM frame to the base frame, shown in Fig. 3. With the quadruped CoM Jacobian, \mathbf{J}_G , tasks relating to it can be integrated into the null space based operational space control framework. The centroidal momentum of the quadruped part \mathbf{h}_G is derived as

$$\mathbf{h}_G = \begin{bmatrix} \mathbf{k}_G \\ \mathbf{l}_G \end{bmatrix} = \mathbf{I}_G \mathbf{v}_G = \mathbf{I}_G \mathbf{J}_G \dot{\mathbf{q}} = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M}_{quad} \dot{\mathbf{q}}, \quad (5)$$

where \mathbf{k}_G and \mathbf{l}_G represent the 3-DOF centroidal angular and translational momentum of the quadruped part respectively. \mathbf{k}_G will be used in section 3.4 to achieve whole-body angular compliance.

2.2 Wheel Kinematic Contact Constraints

Assume that there is no slippage between the contact points and the ground, the translational kinematic wheel contact constraints are as follows,

$$\begin{aligned}\mathbf{v}_c &= \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0}, \\ \dot{\mathbf{v}}_c &= \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0},\end{aligned}\quad (6)$$

where \mathbf{v}_c is the translational velocities of all wheel contact points. Wheel contact points always change, which is different with fixed contact points on the legs of conventional quadruped robots [17]. The velocity and acceleration relationship between the contact point and the origin of the fixed wheel center frame are derived using spatial vectors.

$$\mathbf{v}_{c_i} = \begin{bmatrix} \boldsymbol{\omega}_{c_i} \\ \mathbf{v}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_i \mathbf{v}_i - \boldsymbol{\Phi}_{c_i} \boldsymbol{\omega}_{i_z}, \quad (7a)$$

$$\dot{\mathbf{v}}_{c_i} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_{c_i} \\ \dot{\mathbf{v}}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_i \dot{\mathbf{v}}_i - \dot{\boldsymbol{\Phi}}_{c_i} \boldsymbol{\omega}_{i_z} - ({}^{c_i}\mathbf{X}_i \mathbf{v}_i) \times \boldsymbol{\Phi}_{c_i} \boldsymbol{\omega}_{i_z}, \quad (7b)$$

where \mathbf{v}_i is the spatial velocity of wheel i and \mathbf{v}_{c_i} is the spatial velocity of contact point i . \mathbf{v}_{c_i} and $\boldsymbol{\omega}_{c_i}$ is the translational and angular velocities respectively for the contact point i . ${}^{c_i}\mathbf{X}_i$ transforms spatial vectors from the wheel frame i to the contact point c_i frame (see Fig. 3). $\boldsymbol{\Phi}_{c_i} \in \mathbb{R}^6$ represents the free modes of contact joint i , which is between the contact point and the ground, it depends on the orientation definition of the contact frame. $\boldsymbol{\omega}_{i_z}$ is the angular component in z direction in the spatial velocity \mathbf{v}_i . From the translational velocity \mathbf{v}_{c_i} and acceleration $\dot{\mathbf{v}}_{c_i}$ of (7), we can derive the detailed expression of the kinematic contact constraint in (6), including each contact \mathbf{J}_{c_i} and $\dot{\mathbf{J}}_{c_i} \dot{\mathbf{q}}$,

$$\dot{\mathbf{J}}_{c_i} \dot{\mathbf{q}} = -\mathbf{S}_c ({}^{c_i}\mathbf{X}_i \mathbf{v}_i) \times \boldsymbol{\Phi}_{c_i} \boldsymbol{\omega}_{i_z}, \quad (8)$$

where $\mathbf{S}_c = [\mathbf{0}_{3 \times 3} \ \mathbf{1}_{3 \times 3}]$ is a selection matrix. (8) allows the contact force \mathbf{F}_c to be extracted by joint torque $\boldsymbol{\tau}$ as well as the contact constraints in (6).

$$\mathbf{F}_c = -\bar{\mathbf{J}}_c^T \mathbf{S}^T \boldsymbol{\tau} + \boldsymbol{\mu}_c + \boldsymbol{\rho}_c = \mathbf{A}_c^T \boldsymbol{\tau} + \mathbf{B}_c, \quad (9)$$

where $\bar{\mathbf{J}}_c$ is the dynamically generalized inverse of the contact Jacobian \mathbf{J}_c . $\boldsymbol{\mu}_c$ and $\boldsymbol{\rho}_c$ and task space coriolis and gravity force, and $\boldsymbol{\mu}_c$ is dependent on $\dot{\mathbf{J}}_c \dot{\mathbf{q}}$.

$$\boldsymbol{\mu}_c = \bar{\mathbf{J}}_c^T \mathbf{C} - \boldsymbol{\Lambda}_c \dot{\mathbf{J}}_c \dot{\mathbf{q}}, \quad (10)$$

Then (9) is only dependent on joint torque and can be integrated into one QP solver for torque optimization. Then the relationship extractions for all tasks in following section can combine \mathbf{A}_c and \mathbf{B}_c naturally. From the dynamics model (2) and the contact force (9), the expected joint acceleration $\ddot{\mathbf{q}}$ is written only depending on $\boldsymbol{\tau}$.

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1} [(\mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}) - \mathbf{C} - \mathbf{G}] = \mathbf{A}_q^T \boldsymbol{\tau} + \mathbf{B}_q. \quad (11)$$

2.3 Extractions for Task References in Operational Space

For locomotion and manipulation, assume the tasks are divided into K hierarchies (the smaller level i , the higher priority). To achieve each task with quadratic forms required by the QP solver, the relationship between each task references and the optimization variables should be derived. By multiply the dynamically contact consistent inverse Jacobian at both sides of the generalized dynamics model in (2), the task i acceleration in operational space can be written only depending on the actuated torque $\boldsymbol{\tau}$.

$$\ddot{\mathbf{x}}_i = \boldsymbol{\Lambda}_{i|prec(i)}^{-1} [\tilde{\mathbf{F}}_{i|prec(i)} + \mathbf{T}_{i|prec(i)} - \boldsymbol{\mu}_{i|prec(i)} - \boldsymbol{\rho}_{i|prec(i)}], \quad (12)$$

where $\boldsymbol{\Lambda}_{i|prec(i)}$, $\boldsymbol{\mu}_{i|prec(i)}$, $\boldsymbol{\rho}_{i|prec(i)}$, $\tilde{\mathbf{F}}_{i|prec(i)}$ represent respectively the operational space inertia, Coriolis/centrifugal, gravity forces and operational space force for task i which are in the null space of tasks $\leq i$, subjected as the subscript $|prec(i)$. $\mathbf{T}_{i|prec(i)}$ combines operational space forces of the task lower than level i , detailed in [13]. Each task acceleration $\ddot{\mathbf{x}}_i$ is derived and arranged as

$$\ddot{\mathbf{x}}_i = \mathbf{A}_i^T \boldsymbol{\tau} + \mathbf{B}_i, \quad (13)$$

where

$$\mathbf{A}_i^T = \mathbf{J}_{i|prec(i)} \mathbf{M}^{-1} (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T) + \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|prec(t)}^T \bar{\mathbf{J}}_{t|prec(t)}^T (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T), \quad (14a)$$

$$\mathbf{B}_i = \mathbf{J}_{i|prec(i)} \mathbf{M}^{-1} \mathbf{J}_c^T \mathbf{B}_c + \dot{\mathbf{J}}_i \dot{\mathbf{q}} - \mathbf{J}_i \mathbf{M}^{-1} \{ \mathbf{C} + \mathbf{G} - \sum_{t=1}^{i-1} \mathbf{J}_{t|prec(t)}^T \bar{\mathbf{J}}_{t|prec(t)}^T \mathbf{J}_c^T \mathbf{B}_c \}, \quad (14b)$$

where $\mathbf{J}_{i|prec(i)} = \mathbf{J}_i \mathbf{N}_{prec(i)}$ represent the operational space Jacobian for task $i_{n \times n}$, which are in the null space of previous tasks, represented as $prec(i)$. $\mathbf{N}_{prec(i)}$ is the null space by combining all tasks above level i . In this paper, $\mathbf{N}_{prec(1)} = \mathbf{1}$. From (14), we see that the task i is dependent on previous $i-1$ tasks and needs careful derivations, which is not suitable for fast calculation. Therefore, we derive the following formulas which can be calculated in a recursive way.

$$\mathbf{A}_i^T = \boldsymbol{\Lambda}_{i|prec(i)}^{-1} (\mathbf{A}_{F_i}^T + \mathbf{A}_{T_i}^T), \quad (15a)$$

$$\mathbf{B}_i = \boldsymbol{\Lambda}_{i|prec(i)}^{-1} (\mathbf{B}_{F_i} + \mathbf{B}_{T_i} - \boldsymbol{\eta}_{i|prec(i)}), \quad (15b)$$

where $\boldsymbol{\eta}_{i|prec(i)} = \boldsymbol{\mu}_{i|prec(i)} + \boldsymbol{\rho}_{i|prec(i)}$. \mathbf{A}_i and \mathbf{B}_i can be derived recursively using formulas as follows,

$$\mathbf{A}_{F_i}^T = \bar{\mathbf{J}}_{i|prec(i)}^T (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T) \quad (16a)$$

$$\mathbf{B}_{F_i} = \bar{\mathbf{J}}_{i|prec(i)}^T \mathbf{J}_c^T \mathbf{B}_c \quad (16b)$$

$$\mathbf{A}_{s_i}^T = \mathbf{A}_{s_{p(i)}}^T + \mathbf{J}_{p(i)|prec(p(i))}^T \mathbf{A}_{F_{p(i)}}^T \quad (16c)$$

$$\mathbf{B}_{s_i} = \mathbf{B}_{s_{p(i)}} + \mathbf{J}_{p(i)|prec(p(i))}^T \mathbf{B}_{F_{p(i)}} \quad (16d)$$

$$\mathbf{A}_{T_i}^T = \boldsymbol{\Lambda}_{i|prec(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{A}_{s_i}^T \quad (16e)$$

$$\mathbf{B}_{T_i} = \boldsymbol{\Lambda}_{i|prec(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{B}_{s_i} \quad (16f)$$

$$\mathbf{A}_q^T = \mathbf{M}^{-1} (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T) \quad (16g)$$

$$\mathbf{B}_q = \mathbf{M}^{-1} (\mathbf{J}_c^T \mathbf{B}_c - \mathbf{C} - \mathbf{G}) \quad (16h)$$

$$\mathbf{A}_c^T = -\bar{\mathbf{J}}_c^T \mathbf{S}^T \quad (16i)$$

$$\mathbf{B}_c = \boldsymbol{\mu}_c + \boldsymbol{\rho}_c, \quad (16j)$$

where \mathbf{A}_{F_i} , \mathbf{B}_{F_i} , \mathbf{A}_{T_i} and \mathbf{B}_{T_i} are middle-process variables, and \mathbf{A}_{s_i} , \mathbf{B}_{s_i} are used to allow \mathbf{A}_{T_i} and \mathbf{B}_{T_i} to be calculated in a recursive way. The detailed algorithm is developed and shown in Algorithm. 1.

3 Constrained Convex Problem Formulation

In this section, multiple tasks are integrated into one null space based convex optimization. Although this weighted scheme cannot ensure the strict hierarchies, it can allow important tasks to be achieved with better performance by tuning the weight matrix. In addition, several null space based impedance controllers in section 3.4 can be integrated into this scheme compatibly. Furthermore, when there is not enough DOF for one specific task, feasible motion directions can be selected by decomposition methods.

3.1 Optimization in Operational Space

Assume that the robot has enough DOFs to fulfill all the tasks. For K level tasks, there are K desired references (from $\ddot{\mathbf{x}}_1^*$ to $\ddot{\mathbf{x}}_K^*$). The error between the desired references $\ddot{\mathbf{x}}_i^*$ and the expected ones $\ddot{\mathbf{x}}_i$ can be used as inputs of the optimization solver. We define the desired and expected vectors combining all levels' tasks as follows,

$$\begin{aligned}\ddot{\mathbf{x}}^* &= [(\ddot{\mathbf{x}}_1^*)^T, \dots, (\ddot{\mathbf{x}}_i^*)^T, \dots, (\ddot{\mathbf{x}}_K^*)^T]^T, \\ \ddot{\mathbf{x}} &= [\ddot{\mathbf{x}}_1^T, \dots, \ddot{\mathbf{x}}_i^T, \dots, \ddot{\mathbf{x}}_K^T]^T,\end{aligned}\quad (17)$$

where $\ddot{\mathbf{x}}$ can be written into linear form according to $\boldsymbol{\tau}$.

$$\ddot{\mathbf{x}} = \begin{bmatrix} \ddot{\mathbf{x}}_1 \\ \vdots \\ \ddot{\mathbf{x}}_K \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^T \boldsymbol{\tau} + \mathbf{B}_1 \\ \vdots \\ \mathbf{A}_K^T \boldsymbol{\tau} + \mathbf{B}_K \end{bmatrix} = \mathbf{A}^T \boldsymbol{\tau} + \mathbf{B}, \quad (18)$$

where \mathbf{A}_i and \mathbf{A}_j may have different sizes due to that the DOFs accounted by each task level are different, the same with \mathbf{B}_i and \mathbf{B}_j . The cost function $\mathbf{J}(\boldsymbol{\tau})$ in operational space can be designed as

$$\mathbf{J}(\boldsymbol{\tau}) = \|\ddot{\mathbf{x}}^* - \ddot{\mathbf{x}}\|_{\mathbf{w}}^2 + \varepsilon \|\boldsymbol{\tau}\|^2, \quad (19a)$$

$$= \frac{1}{2} \boldsymbol{\tau}^T (\mathbf{A} \mathbf{A}^T \mathbf{w} + \varepsilon \mathbf{1}) \boldsymbol{\tau} + \boldsymbol{\tau}^T \mathbf{A} \mathbf{w} (\mathbf{B} - \ddot{\mathbf{x}}^*), \quad (19b)$$

$$= \frac{1}{2} \boldsymbol{\tau}^T \mathbf{H} \boldsymbol{\tau} + \boldsymbol{\tau}^T \mathbf{g}, \quad (19c)$$

where \mathbf{w} is the weight matrix. \mathbf{H} and \mathbf{g} denote the symmetric Hessian matrix and the gradient vector. When robot DOF has more DOFs required by tasks, ε is set very small to ensure required $\boldsymbol{\tau}$ small. The simple optimization equation in (19) combines multi-hierarchical tasks and contact constraints in (9). The detailed algorithm to solve the optimization problem in (19) is shown in Algorithm. 1, in which $p(i)$ represents the $i-1$ hierarchy.

When the robot DOF is not enough, for conventional weighted QP without null space projections, feasible motion directions are not easy to be distributed. However, for our null space based theme, this issue can be solved by the eigen-decomposition method (presented in [2]) which can select the feasible motion directions.

$$\mathbf{A}_{i|prec(i)} = \mathbf{U}_{r(i)} \boldsymbol{\Sigma}_{r(i)}^{-1} \mathbf{U}_{r(i)}^T. \quad (20)$$

3.2 Optimization in Joint Space

Another null space based approach is done by transferring the method in (19) to the joint space although it is not very apparent. Each desired reference $\ddot{\mathbf{x}}_i^*$ is used to get the desired operational space force $\tilde{\mathbf{F}}_{i|prec(i)}^*$ and then achieve the desired torque $\boldsymbol{\tau}_{i|prec(i)}^*$ using QR decomposition projector in [14].

$$\boldsymbol{\tau}_{i|prec(i)} = \underbrace{\{(\mathbf{P}_{QR}\mathbf{S}^T)^+ \mathbf{P}_{QR}\}}_{\mathbf{P}} \mathbf{J}_{i|prec(i)}^T \tilde{\mathbf{F}}_{i|prec(i)}, \quad (21)$$

where \mathbf{P}_{QR} depends only on the contact Jacobian and can allow $\boldsymbol{\tau}$ independent on contact force. For each task level, the error between $\boldsymbol{\tau}_{i|prec(i)}^*$ and $\boldsymbol{\tau}_{i|prec(i)}$ can be written as

$$\Delta \boldsymbol{\tau}_{i|prec(i)} = \mathbf{P} \mathbf{J}_{i|prec(i)}^T \boldsymbol{\Lambda}_{i|prec(i)} (\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i) \quad (22a)$$

$$= \mathbf{Q}_{i|prec(i)} (\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i) \quad (22b)$$

$$= \mathbf{A}_{\tau_i}^T \boldsymbol{\tau} + \mathbf{B}_{\tau_i}, \quad (22c)$$

Then the QP solver can be designed, not only to optimize the whole torque combining all levels, but also the torque generated by each task level in a recursive way.

$$\arg \min_{\boldsymbol{\tau}} \sum_{i=1}^K \left\| \sum_{j=1}^i \mathbf{Q}_{j|prec(j)} (\ddot{\mathbf{x}}_j^* - \ddot{\mathbf{x}}_j) \right\|_{\mathbf{w}}^2, \quad (23)$$

where $i = 1$ means that the first level is optimized and $i = 2$ means that the first two levels are optimized together and $i = K$ is to optimize the torque for all levels.

3.3 Common Constraints

The advantage of QP is to optimize several tasks simultaneously and integrate several inequality constraints which can include joint limits, actuation limits, unilateral condition and the friction cone, etc. These constraints are expressed as follows,

$$\begin{cases} \mathbf{q}_{min} \leq \mathbf{q}_{|k+1} = \mathbf{q}_{|k} + \dot{\mathbf{q}}_{|k} \delta t + \frac{1}{2} \ddot{\mathbf{q}}_{|k} \delta t^2 \leq \mathbf{q}_{max} \\ \dot{\mathbf{q}}_{min} \leq \dot{\mathbf{q}}_{|k+1} = \dot{\mathbf{q}}_{|k} + \ddot{\mathbf{q}}_{|k} \delta t \leq \dot{\mathbf{q}}_{max} \\ \boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_{|k} \leq \boldsymbol{\tau}_{max} \\ \mathbf{u}_c \cdot \mathbf{F}_c \geq 0 \\ |\mathbf{t}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c) \\ |\mathbf{b}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c). \end{cases} \quad (24)$$

where $(\mathbf{t}_{c_i}, \mathbf{b}_{c_i}, \mathbf{u}_{c_i})$ represent the contact frame i . $(\mathbf{t}_c, \mathbf{b}_c, \mathbf{u}_c)$ combine n_c contact points.

$$\mathbf{s}_c = \begin{bmatrix} \mathbf{s}_{c_1} \\ \vdots \\ \mathbf{s}_{c_{n_c}} \end{bmatrix}, \quad \mathbf{s} \in \{\mathbf{t}, \mathbf{b}, \mathbf{u}\} \quad (25)$$

where $\mathbf{t}_{c_i} = [1 \ 0 \ 0]^T$, $\mathbf{b}_{c_i} = [0 \ 1 \ 0]^T$ and $\mathbf{u}_{c_i} = [0 \ 0 \ 1]^T$. The joint limits can be expressed by $\boldsymbol{\tau}$ using (11). The unilateral and friction cone can be formulated by $\boldsymbol{\tau}$ using (7) and (9).

3.4 Null space based Compatible Compliance Controllers

The whole control framework is shown in Fig. 4. In the decoupling motion generator, given the CoM velocity and the base angular velocity as well as the legged velocity and configuration, the consistent wheel motion $\dot{\mathbf{q}}_w$ can be derived combining one kinematic velocity constraint in (7a) and the quadruped centroidal momentum constraint in (5). The wheel acceleration $\ddot{\mathbf{q}}_w$ can be derived using the kinematic acceleration constraint in (7b) and the centroidal dynamics constraint which is the derivative of (5). The detailed derivations are referred to our another work [23].

Although the adapted trajectory in the joint space can be achieved, to allow the robot to handle various tasks in the operational space and to validate the efficiency of our method, we control three operational space tasks, including the highest hierarchy for the CoM motion tracking (signed as G), the second hierarchy for the wheel motion (signed as w) and the lowest hierarchy for the motion of the manipulator end effector (signed as m). Therefore, there are $K = 3$ tasks or hierarchies.

In this subsection, null space based impedance controllers are developed which have compatibility with the null space based convex optimization. In other words, impedance controllers are consistent with the task hierarchies.

When tracking the whole-robot motion, CoM is a special point that whole-body linear and angular momentum can be set here naturally. It is more suitable since that CoM frame can be set always parallel to inertia frame. Due to model impreciseness, impedance controllers are added which also allow robustness and compliance. Since angular positions at CoM have no sense, only damping for angular momentum tracking is applied. The output of the impedance controller for the CoM is the force in the operational space which are transferred to acceleration by the inverse of the CoM inertia.

$$\ddot{\mathbf{x}}_G^* = \ddot{\mathbf{x}}_G^d + \begin{bmatrix} \Lambda_{G_\omega}^{-1} \mathbf{D}_{G_\omega} (\mathbf{k}_G^d - \mathbf{k}_G) \\ \Lambda_{G_v}^{-1} \{ \mathbf{K}_{G_v} (\mathbf{p}_G^d - \mathbf{p}_G) + \mathbf{D}_{G_v} (\mathbf{v}_G^d - \mathbf{v}_G) \} \end{bmatrix}, \quad (26)$$

where the operational space inertia Λ_G at CoM is consisted of $[\Lambda_{G_\omega}, \Lambda_{G_{\omega v}}; \Lambda_{G_{v\omega}}, \Lambda_{G_v}]$ in which $\Lambda_{G_{v\omega}}$ and $\Lambda_{G_{\omega v}}$ are coupling items. \mathbf{k}_G is the quadruped angular momentum. \mathbf{p}_G and \mathbf{v}_G represent the translational quadruped CoM position and velocity in the inertial frame.

For wheel locomotion, due to that rolling resistance is not estimated, one impedance controller is built in the wheel joint space to compensate the incorrectness of the contact model. The right side of (2) should minus this impedance torque $\boldsymbol{\tau}_{im_w}$ which should be in the null space of task CoM.

$$\boldsymbol{\tau}_{im_w} = \mathbf{N}_G^T \mathbf{S}_w^T [\mathbf{K}_w (\mathbf{q}_w^d - \mathbf{q}_w) + \mathbf{D}_w (\dot{\mathbf{q}}_w^d - \dot{\mathbf{q}}_w)], \quad (27)$$

where \mathbf{S}_w is the selection matrix which maps the impedance force from dimension \mathbb{R}^{n_w} to \mathbb{R}^n . \mathbf{N}_G represent the null space projector of the task CoM Jacobian, \mathbf{J}_G . Then the contact force \mathbf{F}_c and each \mathbf{B}_{F_i} in (16) should change as

$$\begin{aligned} \mathbf{F}_c &= \mathbf{A}_c^T \boldsymbol{\tau} + \mathbf{B}_c + \bar{\mathbf{J}}_c^T \sum \boldsymbol{\tau}_{im_j}, \\ \mathbf{B}_{F_i} &= \bar{\mathbf{J}}_{i|prec(i)}^T \mathbf{J}_c^T \mathbf{B}_c - \bar{\mathbf{J}}_{i|prec(i)}^T \sum \boldsymbol{\tau}_{im_j}. \end{aligned} \quad (28)$$

where $\sum \boldsymbol{\tau}_{im_j}$ means that several impedance controller can be integrated in this way, including wheel impedance torque in (27), base angular impedance torque in (30) and legged impedance force in (31).

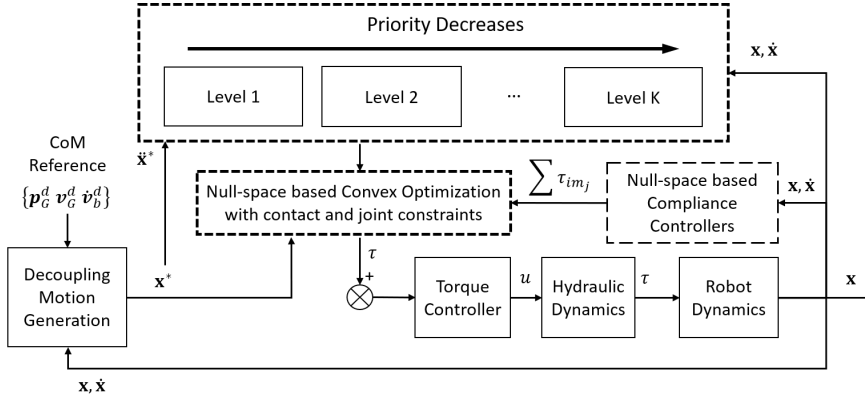


Fig. 4 Whole-body control framework of TowerSIR: given the desired quadruped CoM translational motion and current states, the decoupling motion generator generates legged motion and wheel motion. The null space based convex optimizer outputs the required joint torque to fulfill the motion, in which each task is prioritized and integrated with compliance control to allow robustness. Other impedance controllers $\sum \tau_{imj}$ can be added into the null space based convex optimization.

The manipulator allows the robot to achieve different kinds of missions. To allow it with compliance, one impedance controller is added in the joint space and transferred to the end effector of the manipulator.

$$\ddot{\mathbf{x}}_m^* = \ddot{\mathbf{x}}_m^d + \mathbf{A}_m^T \{ \mathbf{K}_m (\mathbf{q}_m^d - \mathbf{q}_m) + \mathbf{D}_m (\dot{\mathbf{q}}_m^d - \dot{\mathbf{q}}_m) \}. \quad (29)$$

where \mathbf{A}_m and \mathbf{B}_m represent \mathbf{A}_3 and \mathbf{B}_3 in (16) respectively. \mathbf{q}_m is the arm joint position which is achieved by inverse kinematics.

When running on the rough terrain, to allow the base pitch angle with small value, the impedance controller for the base angular motion is designed by

$$\boldsymbol{\tau}_{im_b} = \mathbf{J}_b^T \{ \mathbf{K}_b (\boldsymbol{\theta}_b^d - \boldsymbol{\theta}_b) + \mathbf{D}_b (\boldsymbol{\omega}_b^d - \boldsymbol{\omega}_b) \}, \quad (30)$$

where \mathbf{J}_b is the Jacobian for the base pitch motion, $\boldsymbol{\theta}_b$ and $\boldsymbol{\omega}_b$ represent the base pitch angle and angular velocity.

In addition, to allow legged suspension, another impedance controller for the legs is built as

$$\boldsymbol{\tau}_{im_g} = \mathbf{J}_g^T \{ \mathbf{K}_g (\mathbf{p}_g^d - \mathbf{p}_g) + \mathbf{D}_g (\dot{\mathbf{p}}_g^d - \dot{\mathbf{p}}_g) \}. \quad (31)$$

where $\mathbf{p}_g \in \mathbb{R}^4$ combines all lower-leg-end positions, namely the wheel center position relative to the base frame in forward x direction, \mathbf{J}_g is the relative Jacobian. (30) and (31) should be integrated into the algorithm using (28).

4 Advanced Numerical Simulation Results

The control framework proposed in section 3 is tested on a virtual tetrapod-on-wheel robot with one manipulator, "TowerSIR" (Fig. 1). The QP problem is implemented using the QPOASES library, a real-time open source QP solver [21]. Simulation results are reported using an advanced physical engine (Fig. 5). Our controller uses virtual sensors, including GPS for the real-time robot position, IMU for the base orientation and angular velocities, joint position encoders as well as torque sensors for joint state measurements.

Algorithm 1 Null space based Convex Optimization

Inputs: There are K hierarchies, $\mathbf{M}, \mathbf{C}, \mathbf{G}, \mathbf{q}, \dot{\mathbf{q}}$

$$\mathbf{A}_c^T = -\bar{\mathbf{J}}_c^T \mathbf{S}^T \text{ and } \mathbf{B}_c = \boldsymbol{\mu}_c + \boldsymbol{\rho}_c$$

$$\mathbf{B}_c + = \bar{\mathbf{J}}_c^T \sum \boldsymbol{\tau}_{im_j} \text{ (when impedance controllers should be integrated into the hierarchies.)}$$

$$\bar{\mathbf{x}}_i^*, \mathbf{J}_i, \bar{\mathbf{J}}_i \dot{\mathbf{q}}, \mathbf{J}_{i|prec(i)}, \mathbf{N}_{prec(i)}, \boldsymbol{\mu}_{i|prec(i)}, \boldsymbol{\rho}_{i|prec(i)}, i \in \{1, \dots, K\}$$

Initialization:

$$\mathbf{A}_{s_1} = \mathbf{0}, \mathbf{B}_{s_1} = \mathbf{0}, \mathbf{A}_{T_1} = \mathbf{0} \text{ and } \mathbf{B}_{T_1} = \mathbf{0}.$$

for $i=1$ to K do

$$\mathbf{A}_{F_i}^T = \bar{\mathbf{J}}_{i|prec(i)}^T (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T)$$

$$\mathbf{B}_{F_i} = \bar{\mathbf{J}}_{i|prec(i)}^T \mathbf{J}_c^T \mathbf{B}_c$$

$$\mathbf{B}_{F_i} - = \bar{\mathbf{J}}_{i|prec(i)}^T \sum \boldsymbol{\tau}_{im_j} \text{ (when impedance controllers should be integrated into the hierarchies.)}$$

if $i \neq 1$ then

$$\mathbf{A}_{s_i}^T = \mathbf{A}_{s_{p(i)}}^T + \mathbf{J}_{p(i)|prec(p(i))}^T \mathbf{A}_{F_{p(i)}}^T$$

$$\mathbf{B}_{s_i} = \mathbf{B}_{s_{p(i)}} + \mathbf{J}_{p(i)|prec(p(i))}^T \mathbf{B}_{F_{p(i)}}$$

$$\mathbf{A}_{T_i}^T = \boldsymbol{\Lambda}_{i|prec(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{A}_{s_i}^T$$

$$\mathbf{B}_{T_i} = \boldsymbol{\Lambda}_{i|prec(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{B}_{s_i}$$

end

$$\mathbf{A}_i^T = \boldsymbol{\Lambda}_{i|prec(i)}^{-1} (\mathbf{A}_{F_i}^T + \mathbf{A}_{T_i}^T)$$

$$\mathbf{B}_i = \boldsymbol{\Lambda}_{i|prec(i)}^{-1} (\mathbf{B}_{F_i} + \mathbf{B}_{T_i} - \boldsymbol{\eta}_{i|prec(i)})$$

end

$$\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_i, \dots, \mathbf{A}_K] \text{ and } \mathbf{B} = [\mathbf{B}_1^T, \dots, \mathbf{B}_i^T, \dots, \mathbf{B}_K^T]^T$$

$$\mathbf{H} = \mathbf{A} \mathbf{A}^T \mathbf{w} + \varepsilon \mathbf{1} \text{ and } \mathbf{g} = \mathbf{A} \mathbf{w} (\mathbf{B} - \bar{\mathbf{x}}^*)$$

$$\arg \min_{\boldsymbol{\tau}} \frac{1}{2} \boldsymbol{\tau}^T \mathbf{H} \boldsymbol{\tau} + \boldsymbol{\tau}^T \mathbf{g}$$

s.t. inequality constraints in (24) and other user-defined constraints

The trajectory generation for the CoM motion G and the manipulator motion m are in the operational space, for the wheel motion w is in the joint space. The desired manipulator end-effector trajectory is generated in the base frame using the Kinematics-Dynamics Library (KDL) [20]. It is noticed that conventional acceleration at the end effector should be modified to the spatial acceleration [16].

4.1 Simulation Scenario Description

Although our tetrapod-on-wheel robot can work in various scenarios, we focus on two complex missions, including opening a door when running on a flat ground, and washing a wall when running on uneven terrain.

In order to evaluate the performance of our controller, we perform two simulations using ROS-GAZEBO at the velocity $1m.s^{-1}$. Legged motion (e.g. walking) is not considered in this paper which will be analyzed in future works. Therefore, the whole-body CoM motion is only contributed by the wheel motion. The robot reference is set as a straight line in the robot forward direction. Although the importance levels of the tasks should be decreased from the CoM task to the wheel task, then to the manipulator task, in the two simulations, the weighted matrix is tuned to be identity in the cost function.

In the first scenario, the robot runs on the flat terrain shown in Fig. 5(a). In the process, the robot uses its arm to push a door open which begins at $t \approx 7s$ (see Fig. 6(1c)). The end effector keeps touch with the door for a while and changes its orientation according to the motion of the door.

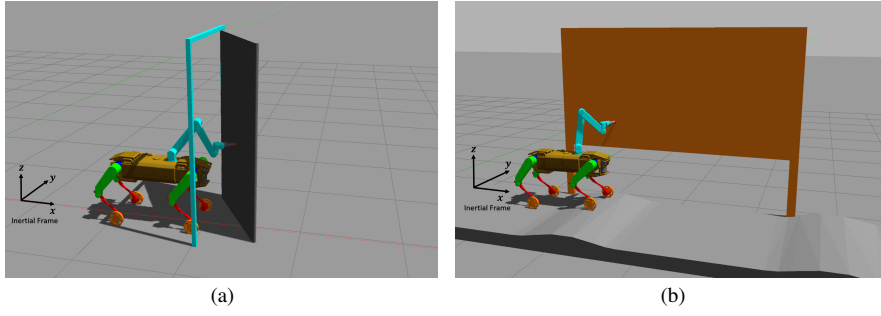


Fig. 5 Two simulation scenarios under GAZEBO simulator. (a) The robot runs along with pushing a door open, including the extension and retraction of the arm. (b) The robot runs on the rough terrain along with washing a wall, including the suspension of the legs and the motion of the arm.

The second scenario relates to a more complex environment in which the robot runs on an uneven terrain (as can be seen in Fig. 5(b)), along with washing a wall using the manipulator end effector which keeps contact with the wall. The robot begins to climb on two bumps at $t \approx 9.8s$ and $t \approx 14.4s$ (see Fig. 6(2a) and (2b)). The arm end effector begins to touch the wall at $t \approx 9.6s$ and ends at $t \approx 14.6s$ as noticed in Fig. 6(2c).

4.2 Result Discussion

The first simulation results are shown in Fig. 6. Along the whole process, the CoM and wheel tracking errors are very small, and the desired and actual trajectories are almost coincident as can be noticed in Fig. 6(1a) and (1b). For the manipulator motion, due to the existence of a virtual spring at the door axis, the position and orientation errors of the manipulator end effector experiences a relatively big fluctuation along the open-door process in Fig. 6(1c), which depends on the weighted matrix in (19) and the stiffness and damping values in the null space based impedance controllers in (28), as well as the door resistance force. With our controller, the robot successfully finish the mission with compliance.

The second simulation results for the tasks are also shown in Fig. 6 with index 2. Because of the existence of two bumps on the rough terrain, compared with the first simulation results, the CoM tracking errors in forward and vertical directions experience a little more errors (see Fig. 6 (2a)), however, the errors are still very small along the whole process. We can see that the wheel tracking errors (see Fig. 6 (2b)) are relatively bigger than those provided when running on the flat ground. Since we assume that the contact points are always at the lowest points of the wheels, it is not precise when the robot runs on slopes, which influence the outputs of the motion generator. This issue will be treated in our future works by one estimation method. To maintain the contact condition between the manipulator end effector and the wall, the manipulator trajectory is defined in a bigger range and the end effector experiences a little higher error along the wall-washing motion (see Fig. 6(2c)). With our null space based convex optimization and the compatible impedance controllers, the robot can still achieve this mission with compliance and efficiency.

To summarize, the results provided by our controller are quite satisfactory. This controller allow the robot to handle complex tasks respecting contact constraints and actuator limits. Even with changing environment and unknown external forces, using the embedded compatible impedance controllers, the robot can successfully finish the missions with compliance.

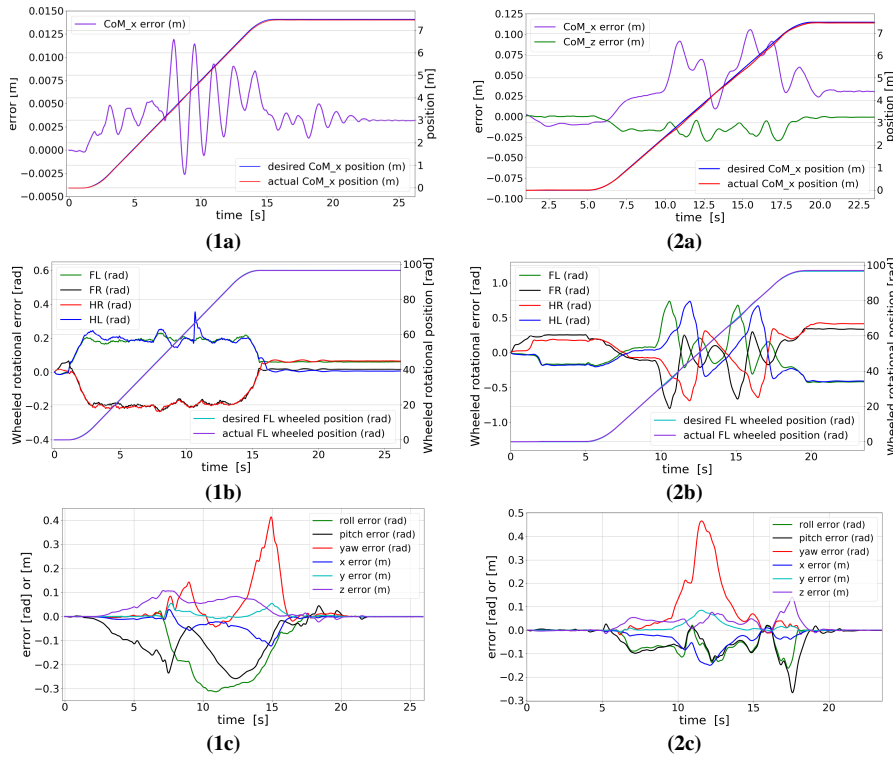


Fig. 6 Monitor of robot states for two simulation scenarios. Index 1 and 2 represent the first and second simulation results respectively. **(1a)** Monitor of quadruped CoM motion in forward x direction. **(1b)** Monitor of the four wheels' rotational position errors: F, L, R, H represent front, left, right and hind. Four wheel motion experience the almost coincident trajectory so that only the desired and actual FL wheel positions are plotted. **(1c)** Monitor of the motion of the arm end effector: desired and actual orientations (roll, pitch, yaw) and positions (x , y , z) of the end effector are relative to the base frame. **(2a)** Monitor of quadruped CoM motion in forward x and vertical z direction. The desired position in z direction is $0.4735m$. **(2b)** Monitor of the four wheels' rotational position error on the rough terrain and only the desired and actual FL wheel positions are plotted. **(2c)** Monitor of the motion of the arm end effector: desired and actual orientations (roll, pitch, yaw) and positions (x , y , z) of the end effector are relative to the base frame.

5 Conclusions

In this paper, we develop a null space based compact form torque controller dedicated to a high-DOF tetrapod-on-wheel robot with one manipulator (Tower/SIR, new mobile robot designed in ISIR laboratory). The control problem is formulated as a Linearly Constrained Quadratic Programming (QP), in which the compact relationships between multiple task references and the actuated torque are extracted using null space based inverse dynamics. The wheel contact model for the tetrapod-on-wheel robot is built using spatial vectors and the wheel contact force is integrated in each task formulation. The cost function is built for several operational space tasks, which can also be transformed to the joint space. Several null space based impedance controllers are integrated into the compact relationships compatibly for achieving task references and compensate model impreciseness, especially the wheel rolling model. Simulation results show that this controller allows the tetrapod-on-wheel

robot with abundant versatility and efficiency of dynamics behaviors, and the robot can handle complex missions with unknown external forces efficiently and compliantly.

The future work is to combine legged and wheel motion together to achieve more complex tasks, e.g. obstacle clearance, jump motion, big push recovery. One online legged motion generator will be developed using whole-body dynamics and model based predictive control.

References

1. Khatib O. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 1987, 3(1): 43-53.
2. Sentis L, Khatib O. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2005, 2(04): 505-518.
3. Bouyarmane K, Kheddar A. Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE*, 2011: 4414-4419.
4. Salini J, Padois V, Bidaud P. Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions. *Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE*, 2011: 1283-1290.
5. Ponce O E R. Generation of the whole-body motion for humanoid robots with the complete dynamics. *Universite Toulouse III Paul Sabatier*, 2014.
6. Kanoun O, Lamiroux F, Wieber P B. Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 2011, 27(4): 785-792.
7. De Lasa M, Mordatch I, Hertzmann A. Feature-based locomotion controllers. *ACM Transactions on Graphics (TOG). ACM*, 2010, 29(4): 131.
8. Escande A, Mansard N, Wieber P B. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 2014, 33(7): 1006-1028.
9. Wensing P M. Optimization and control of dynamic humanoid running and jumping[D]. *The Ohio State University*, 2014.
10. Wensing P M, Orin D E. Generation of dynamic humanoid behaviors through task-space control with conic optimization. *Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE*, 2013: 3103-3109.
11. Mastalli C, Havoutis I, Focchi M, et al. Hierarchical planning of dynamic movements without scheduled contact sequences. *Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE*, 2016: 4636-4641.
12. Focchi M, Del Prete A, Havoutis I, et al. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots*, 2017, 41(1): 259-272.
13. Sentis L. Synthesis and control of whole-body behaviors in humanoid systems. 2007.
14. Righetti L, Buchli J, Mistry M, et al. Inverse dynamics control of floating-base robots with external constraints: A unified view. *Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE*, 2011: 1085-1090.
15. Mistry M, Nakanishi J, Cheng G, et al. Inverse kinematics with floating base and constraints for full body humanoid robot control. *Humanoids 2008. 8th IEEE-RAS International Conference on. IEEE*, 2008: 22-27.
16. Featherstone, Roy. *Rigid body dynamics algorithms*. Springer, 2014.
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. Bellicoso C D, Gehring C, Hwangbo J, et al. Perception-less terrain adaptation through whole body control and hierarchical optimization[C]//*Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on. IEEE*, 2016: 558-564.
19. Semini C, Barasuol V, Goldsmith J, et al. Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max. *IEEE/ASME Transactions on Mechatronics*, 2017, 22(2): 635-646.
20. Smits, R. KDL: Kinematics and Dynamics Library, <http://www.oroocos.org/kdl>
21. Ferre H J, Kirches C, Potschka A, et al. qpOASES: A parametric active-set algorithm for quadratic programming[J]. *Mathematical Programming Computation*, 2014, 6(4): 327-363.
22. Walker M W, Orin D E. Efficient dynamic computer simulation of robotic mechanisms[J]. *Journal of Dynamic Systems, Measurement, and Control*, 1982, 104(3): 205-211.
23. Wenqian DU, Faiz Benamar. Development of a Compliant Motion Generator for a Tetrapod-on-Wheel Robot using Integrated Kinematics and Centroidal Dynamics. (Submitted)