



**HAL**  
open science

# Motion generation of four-limb robots using whole-body torque control

Wenqian Du

► **To cite this version:**

Wenqian Du. Motion generation of four-limb robots using whole-body torque control. Robotics [cs.RO]. Sorbonne Université, 2020. English. NNT : 2020SORUS067 . tel-03191313

**HAL Id: tel-03191313**

**<https://theses.hal.science/tel-03191313v1>**

Submitted on 7 Apr 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.



École doctorale n° 391 : Sciences mécaniques, acoustique, électronique et robotique de Paris

## THÈSE DE DOCTORAT

*présentée à*

**Institut des Systèmes Intelligents et de Robotique: ISIR**

*par*

**Wenqian DU**

*pour obtenir le diplôme de docteur délivré par*

**SORBONNE UNIVERSITÉ**

Spécialité: Robotique

### **Motion Generation of Four-limb Robots using Whole-body Torque Control**

Directeur de thèse : Faïz BENAMAR

*soutenance prévue le Novembre 24, 2020*

**Jury:**

Olivier STASSE	- Directeur de Recherche au CNRS	Rapporteur
Christine CHEVALLEREAU	- Directrice de Recherche au CNRS	Rapporteur
Stéphane DONCIEUX	- Professeur à Sorbonne Université	Examinator
Pierre-Brice WIEBER	- Chargé de Recherche à INRIA Grenoble	Examinator
Christian OTT	- Department Head of German Aerospace Center (DLR)	Examinator
Faïz BENAMAR	- Professeur à Sorbonne Université	Directeur de thèse





# Abstract:

The thesis presents the whole-body motion generation of two four-limb robots using whole-body torque control, including the rolling-mode generation of one quadruped-on-wheel robot, “Towr*ISIR*”, and the trotting-mode generation of one quadruped robot with parallelogram mechanisms, “Quad*ISIR*”. Both the wheeled and legged motion generators are developed by using the robot generalized dynamics and centroidal dynamics models, and they are verified by our improved proposed torque controllers.

Initially, we propose the concept called prioritized impedance controller. We integrate it into four kinds of improved hierarchical operational-space torque control frameworks: one pseudo-inverse based dynamics method and three improved optimization-based techniques, including one null-space based weighted optimization solver, one hierarchical optimization solver, and one hybrid hierarchical-weighted solver. The integrated prioritized impedance controller enables various impedance control forces to be consistent and compatible with multi-task hierarchies. The concept can handle more general impedance controllers, including the ones built directly on specific designed tasks, and ones indirectly acting on the whole-body behaviors. For the pseudo-inverse based dynamics method and the null-space based weighted scheme, each task acceleration with the prioritized impedance controller is extracted using one proposed modified dynamics model. In contrast with the conventional weighted scheme, we propose to use null-space projections among the weighted subtasks. Then another new dynamics model is proposed for the hierarchical optimization solver and the hybrid hierarchical-weighted control scheme, it is used to extract each task acceleration with the quadratic form depending on the prioritized impedance controller. One of the four torque controllers can be selected depending on application requirements, and they are tested in several simulation scenarios using the Towr*ISIR* with one manipulator on its floating base.

Then, we propose two rolling-mode motion generators to track the centroidal motion of Towr*ISIR*. The whole-body inverse kinematics model is derived using spatial vectors. By given the desired legged-suspension motion, the wheel motion is extracted out mathematically depending on the base and the legged motions, then the first 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 second whole-body motion generator is more general which enables the robot ability to cross various rough terrains with much-altitude difference. The generator is also used to track the whole-robot centroidal motion reference, including the centroidal translational motion in the forward and lateral directions in the inertial frame, and the centroidal height motion w.r.t. the base frame in the base vertical direction. Then the whole-body motion generator is developed by combining both the kinematics model, the wheel-center motion model, the robot centroidal momentum and dynamics models, and one proposed altitude control model. Finally, the two adaptive motion generators are verified, and Towr*ISIR* is simulated to be driven on various rough terrains with compatible dynamic behaviors by using our proposed compliant torque controller.

Finally, we propose a novel legged locomotion principle for tracking the base spatial motion of one quadruped robot, Quad*ISIR*, which can generate legged motion automatically. The principle is based on a new legged locomotion generator for a quadruped robot using the centroidal properties without massless-leg assumption. Our quadruped robot is new designed with symmetric legged dynamics properties using parallelogram mechanisms, and the relative generalized dynamics model is built. The centroidal momentum and dynamics models are established by combining the base and each leg centroidal properties, in which the virtual joint between the base and each-leg CoM is used instead of each real joint. This setting is advantageous for the proposed locomotion principle and the legged motion generator. The stance legs follow the base motion without conflicting the contact constraints. The swing legs compensate the delay influences (to centroidal motion) by the stance legs. The adaptive swing-leg motion is achieved in the running process without pre-defined trajectories. We use the contact-constraint consistent multi-task torque controller to enable the robot with compliant dynamics properties. Quad*ISIR* is simulated to walk, trot, turn, and recover from big lateral push force on the base, to verify the usefulness and validity of our new locomotion principle based legged motion generator.

**Keywords:** Four-limb Robot; Generalized dynamics; Operational space dynamics; Whole-body kinematics; Prioritized impedance controller; Hierarchical torque control; Wheel-motion generation; Legged-motion generation.

# List of Abbreviations

<b>COM</b>	Center of mass
<b>COP</b>	center of pressure
<b>CPG</b>	Central pattern generator
<b>DOF</b>	Degrees of freedom
<b>EOM</b>	Equations of motion
<b>FL</b>	Front-left leg
<b>FR</b>	Front-right leg
<b>HL</b>	Hind-left leg
<b>HR</b>	Hind-right leg
<b>OSC</b>	Operational space control
<b>P-OSC</b>	Prioritized operational space control
<b>ROS</b>	Robot operating system
<b>IMU</b>	Inertial measurement unit
<b>GPS</b>	Global positioning system
<b>LQR</b>	Linear quadratic regulator
<b>QP</b>	Quadratic programming
<b>PIC</b>	Prioritized impedance controller
<b>P-WQP</b>	PIC based weighted QP torque solver
<b>P-HQP</b>	PIC based hierarchical QP torque solver
<b>P-HWQP</b>	PIC based hierarchical and prioritized-weighted QP torque solver
<b>SLIP</b>	Spring loaded inverted pendulum
<b>ISIR</b>	Institut des Systèmes Intelligents et de Robotique
<b>Towr<i>ISIR</i></b>	Tetrapod-on-wheel Robot designed in ISIR
<b>Quad<i>ISIR</i></b>	Quadruped robot designed in ISIR
<b>HAA</b>	Hip abduction & adduction joint/link
<b>HFE</b>	Hip flexion & extension joint/link
<b>KFE</b>	Knee flexion & extension joint/link
<b>WFB</b>	Wheel forward & backward joint/link for Towr <i>ISIR</i>
<b>Ankle</b>	Ankle joint/link for Quad <i>ISIR</i>
<b>TO</b>	Trajectory optimization
w.r.t.	With respect to
s.t	Subject to

# List of Symbols

## Common Variables Throughout this Thesis:

$M$	Generalized inertia matrix
$C$	Generalized Centripetal and Coriolis force
$G$	Generalized gravity force
$q$	Generalized joint configuration
$S$	Selection matrix of the actuated torque
$S_u$	Selection matrix for the head three components of one 6D vector
$S_c$	Selection matrix for the tail three components of one 6D vector
${}^B\mathbf{X}_A$	Transformation matrix for spatial vectors from frame A to frame B
${}^B\mathbf{R}_A$	Rotation matrix for 3D vectors from frame A to frame B
${}^B\mathbf{T}_A$	Homogeneous transformation matrix from frame A to frame B
${}^B\mathbf{p}_A$	Point A position expressed in frame B
$\mathbf{J}_{c_i}$	Jacobian for the contact $i$ translational velocity
$\mathbf{J}_c$	Jacobian relating to all contact-point translational velocities
$\mathbf{J}_G$	Jacobian relating to the centroidal motion
$\mathbf{v}_{c_i}$	Contact-point $i$ 6-dimensional spatial velocity
$\mathbf{v}_{c_i}$	Contact-point $i$ 3-dimensional translational velocity
$\boldsymbol{\omega}_{c_i}$	Contact-point $i$ 3-dimensional angular velocity
$\mathbf{v}_c$	All contact-point translational velocity
$\mathbf{x}_b$	Floating-base position and orientation
$\mathbf{p}_b$	Floating-base position in the inertial frame
$\boldsymbol{\theta}_b$	Floating-base orientation angle
$\mathbf{v}_b$	Floating-base 6-dimensional spatial velocity
$\mathbf{v}_b$	Floating-base 3-dimensional translational velocity
$\boldsymbol{\omega}_b$	Floating-base 3-dimensional translational velocity
$\mathbf{v}_G$	Centroidal 6-dimensional spatial velocity
$\mathbf{v}_G$	Centroidal translational velocity
$\boldsymbol{\omega}_G$	Centroidal angular velocity
$\mathbf{I}_G$	Composite rigid-body inertia at the robot CoM
$\mathbf{I}_b^C$	Composite rigid-body inertia at the base
$\mathbf{I}_b$	Base spatial inertia
$\Phi_b$	Free-mode matrix of the floating-base joint
$\mathbf{h}_G$	Centroidal momentum
$\mathbf{p}_G$	Centroidal position
$\tilde{\mathbf{p}}_G$	Centroidal position in the base frame
$\mathbf{k}_G$	Centroidal translational momentum

$l_G$	Centroidal angular momentum
$N_c$	Null-space projector for the contact constraint
$\bar{J}$	Dynamically inverse of any Jacobian
$F_c$	Operational force at the contact points
$\mu_c$	Operational Coriolis and centripetal force at the contact points
$\rho_c$	Operational gravity force at the contact points
$\Lambda_c$	Operational inertia at the contact points
$K$	Stiffness matrix
$D$	Damping matrix
$\delta t$	Fixed sampling time
$\bar{t}$	Normalized time

### Specific Variables for System Modeling:

$i, j$	Subscript of the body or joint index, different with the meanings in other chapters
$k$	Subscript of the hierarchy index, different with the meanings in other chapters
$\Phi_i$	Free-mode matrix of the generalized joint $j$
$\mathbf{v}_i$	Body $j$ 6-dimensional spatial velocity
$\mathbf{v}_i$	Body $j$ 3-dimensional translational velocity
$\boldsymbol{\omega}_i$	Body $j$ 3-dimensional angular velocity
$\mathbf{a}_i$	Body $j$ spatial acceleration, equal to $\dot{\mathbf{v}}_i$
$\mathbf{i}_i$	Jacobian for body $j$ spatial velocity
$\mathbf{I}_i$	Body $j$ spatial inertia
${}^j\mathbf{I}_i$	Spatial inertia of body $i$ , expressed in frame $j$
$\bar{\mathbf{I}}_i$	Body $i$ angular inertia
$m_i$	Body $i$ mass
$m_G$	Robot mass
$\mathbf{f}_i$	Spatial net force applied on body $i$
${}^j\mathbf{f}_i$	Spatial net force applied on body $i$ , expressed in frame $j$
$\mathbf{h}_i$	Body $i$ spatial momentum
$l_i$	Body $i$ translational momentum
$\mathbf{k}_i$	Body $i$ angular momentum
$\mathbf{J}_i$	Body $i$ motion Jacobian
$\mathbf{J}_k$	Hierarchy $k$ task Jacobian
$\mathbf{P}_{direct}$	Contact-constraint consistent direct projector
$\mathbf{P}_{osc}$	Contact-constraint consistent operational-space projector
$\mathbf{P}_{QR}$	Contact-constraint consistent QR-decomposition projector
$\mathbf{A}_G$	Centroidal momentum matrix (CMM)
$\mathbf{J}_{k pre(k)}$	Hierarchy $i$ task Jacobian projected in the null-spaces of previous hierarchical tasks
$\mathbf{N}_{pre(k)}$	Null space combining all the previous hierarchical tasks of hierarchy $k$
$\Lambda_{k pre(k)}$	Operational-space inertia for hierarchy $k$ projected in null-spaces of previous hierarchies

---

$\boldsymbol{\mu}_{k pre(k)}$	Operational-space Centripetal and Coriolis force for hierarchy $k$ projected in null-spaces of previous hierarchies
$\boldsymbol{\rho}_{k pre(k)}$	Operational-space gravity force for hierarchy $k$ projected in null-spaces of previous hierarchical tasks
$\mathbf{F}_{k pre(k)}$	Operational-space force for hierarchy $k$ projected in null-spaces of previous hierarchies
$\mathbf{T}_{k pre(k)}$	Combination of all previous hierarchical operational-space forces, projected in the hierarchy $k$ operational-space force

### Specific Variables for Torque Control for Locomotion Robots:

$Q(\boldsymbol{\tau})$	Cost function with the optimization variable $\boldsymbol{\tau}$ in P-WQP
$Q_i(\boldsymbol{\tau}_i)$	Cost function with the optimization variable $\boldsymbol{\tau}_i$ in P-HQP or P-HWQP
$\varepsilon$	Small number ensuring positive definiteness of the objective Hessian
$\boldsymbol{\mu}_c$	Normal direction vector of contact points
$\mathbf{t}_c$	Tangential direction vector of contact points
$\mathbf{b}_c$	Lateral direction vector of contact points
$\mu$	Ground friction coefficient
$\ddot{\mathbf{x}}_{i,j}$	Subtask $j$ acceleration in hierarchy $i$ in P-HWQP
$\ddot{\mathbf{x}}_i$	Hierarchy $i$ task acceleration in P-WQP, P-HQP or P-HWQP
$\ddot{\mathbf{x}}$	Combination of each hierarchy task acceleration $\ddot{\mathbf{x}}_i$ in P-WQP
$\boldsymbol{\tau}$	Optimization variable in P-WQP
$\boldsymbol{\tau}_i$	Optimization variable for hierarchy $i$ in P-HQP or P-HWQP
$\boldsymbol{\tau}_{act_i}$	Required torque for hierarchy $i$ in P-HQP or P-HWQP
$\mathbf{A}_c, \mathbf{B}_c$	Variables of the quadratic-form contact force depending on $\boldsymbol{\tau}$ in P-WQP
$\mathbf{A}_q, \mathbf{B}_q$	Variables of the quadratic-form joint acceleration depending on $\boldsymbol{\tau}$ in P-WQP
$\mathbf{A}_{c_i}, \mathbf{B}_{c_i}$	Variables of the quadratic-form contact force in hierarchy $i$ depending on $\boldsymbol{\tau}_i$ in P-HQP or P-HWQP
$\mathbf{A}_{q_i}, \mathbf{B}_{q_i}$	Variables of the quadratic-form joint acceleration in hierarchy $i$ depending on $\boldsymbol{\tau}_i$ in P-HQP or P-HWQP
$\mathbf{A}_i, \mathbf{B}_i$	Variables of the quadratic-form $\ddot{\mathbf{x}}_i$ depending on $\boldsymbol{\tau}$ in P-WQP, depending on $\boldsymbol{\tau}_i$ in P-HQP or P-HWQP
$\mathbf{A}_{i_j}, \mathbf{B}_{i_j}$	Variables of the quadratic-form $\ddot{\mathbf{x}}_{i_j}$ depending on $\boldsymbol{\tau}_i$ in P-HWQP
$\mathbf{J}_{i_j}$	Hierarchy $i$ subtask $j$ Jacobian, in P-HWQP
$\mathbf{J}_{i_j pre(i_j)}$	Hierarchy $i$ subtask $j$ Jacobian projected in the null-spaces of previous hierarchies, and in the null-spaces of the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP
$\mathbf{N}_{pre(i_j)}$	Null space combining all the previous $i - 1$ hierarchies, and the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP
$\boldsymbol{\Lambda}_{i_j pre(i_j)}$	Operational-space inertia for hierarchy $i$ subtask $j$ projected in the null-spaces of previous hierarchies, and the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP
$\boldsymbol{\mu}_{i_j pre(i_j)}$	Operational-space Centripetal and Coriolis force for hierarchy $i$ subtask $j$ projected in the null-spaces of previous hierarchies, and the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP
$\boldsymbol{\rho}_{i_j pre(i_j)}$	Operational-space gravity force for hierarchy $i$ subtask $j$ projected in the null-spaces of previous hierarchies, and the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP

$\mathbf{F}_{i_j pre(i_j)}$	Operational-space force for hierarchy $i$ subtask $j$ projected in the null-spaces of previous hierarchies, and the weighted $j - 1$ subtasks in hierarchy $i$ , in P-HWQP
$\mathbf{T}_{i_j pre(i_j)}$	Combination of operational-space forces in the previous hierarchies and the weighted $j - 1$ subtasks in hierarchy $i$ , projected in the hierarchy $i$ subtask $j$ operational-space force, in P-HWQP
$\tau_{imk}$	Feedback impedance torque with index $k$
$\Gamma$	Sum of all feedback impedance controllers by respecting task hierarchy

### Specific Variables for Wheel-motion Generation for Towr*ISIR*:

$i$	Subscript of the leg index
$\mathbf{J}_{k_i}$	Jacobian relating to the knee (KFE) joint velocity
$\mathbf{v}_{w_i}$	Wheel-body $i$ spatial velocity
$\mathbf{v}_{k_i}$	Knee-body $i$ spatial velocity
$\mathbf{q}_{w_i}$	Wheel $i$ joint position
$\mathbf{q}_w$	all wheel joint position
$\mathbf{q}_{r_i}$	Leg $i$ revolute joint positions, except the leg $i$ wheel joint
$\mathbf{q}_r$	Combination of all leg-revolute joint position $\mathbf{q}_{r_i}$ , except the wheel joints
$\mathbf{q}_{g_i}$	Combination of leg $i$ revolute joint and leg $i$ wheel joint configurations
$\mathbf{q}_g$	Combination of all leg joint configurations
$\Phi_{c_i}$	Free mode matrix of the virtual contact joint
$\Phi_{w_i}$	Free mode matrix of the leg $i$ wheel joint
$\mathbf{I}_{G_w}$	Left three columns of $\mathbf{I}_G$
$\mathbf{I}_{G_v}$	Right three columns of $\mathbf{I}_G$
$\mathbf{I}_{w_i}$	Wheel $i$ spatial inertia
$\mathbf{I}_{b_w}$	Left three columns of $\mathbf{I}_b$
$\mathbf{I}_{b_v}$	Right three columns of $\mathbf{I}_b$
$\Upsilon$	Outputs of the whole-body motion generator when the robot rolls on the rough terrain

### Specific Variables for Legged Motion Generation for Quad*ISIR*:

$i$	Subscript of the leg index
$\underline{\mathbf{M}}$	Generalized inertia matrix derived using minimum DOFs
$\underline{\mathbf{C}}$	Generalized Centripetal and Coriolis force using minimum DOFs
$\underline{\mathbf{G}}$	Generalized gravity force using minimum DOFs
$\mathbf{M}_i$	Generalized inertia matrix derived assuming only leg- $i$ body inertias are nonzero
$\mathbf{C}_i$	Generalized Centripetal and Coriolis force assuming only leg- $i$ body inertias are nonzero
$\mathbf{R}_{i_j}$	Rotation matrix of the leg $i$ joint $j$
$\lambda$	Coefficient matrix between each leg CoM and foot in HAA frame
$\lambda_i$	Coefficient matrix between leg $i$ CoM and foot in HAA frame
$\mathbf{q}_{r_i}$	Leg $i$ joint configuration
$\mathbf{q}_{a_i}$	Leg $i$ actuated-joint configuration

---

$\tilde{\mathbf{p}}_i$	Leg $i$ CoM position in the base frame
$\tilde{\mathbf{p}}_{sw_i}$	Swing leg $i$ CoM position
$\tilde{\mathbf{p}}_{st_j}$	Stance leg $j$ CoM position
$\mathbf{v}_i$	Leg $i$ CoM spatial velocity expressed in the leg $i$ CoM frame
$\mathbf{v}_i$	Leg $i$ CoM translational velocity expressed in the leg $i$ CoM frame
$\boldsymbol{\omega}_i$	Leg $i$ CoM angular velocity expressed in the leg $i$ CoM frame
$\tilde{\mathbf{v}}_i$	Leg $i$ CoM translational velocity w.r.t. the base expressed in leg $i$ CoM frame
$\tilde{\boldsymbol{\omega}}_i$	Leg $i$ CoM angular velocity w.r.t. the base expressed in leg $i$ CoM frame
$\mathbf{I}_i$	Composite leg $i$ CoM spatial inertia
$\bar{\mathbf{I}}_i$	Composite leg $i$ CoM angular inertia
$\bar{\mathbf{I}}_G$	Composite rigid-body angular inertia at the robot CoM
$\bar{\mathbf{I}}_b^C$	Composite rigid-body angular inertia at the base
$\bar{\mathbf{I}}_b$	Base angular inertia
${}^i\mathbf{X}_b$	Transformation matrix from base frame to leg $i$ CoM frame
${}^i\mathbf{X}_G$	Transformation matrix from robot CoM frame to leg $i$ CoM frame
$m$	One leg mass
$\tilde{\mathbf{p}}_{c_i}$	Leg $i$ foot position in the base frame
$\mathbf{h}_b$	Contributed centroidal momentum by the base motion
$\mathbf{h}_i$	Contributed centroidal momentum by the leg $i$ CoM motion
$\mathbf{J}_m$	Jacobian relating to the motion of passive Ankle joints
$\mathbf{N}_m$	Null-space projection relating to the parallel-mechanism constraints
$\mathbf{J}_a$	Jacobian relating to the motion of actuated joints
$\mathbf{J}_i$	Jacobian relating to the leg $i$ CoM motion
$\mathbf{J}_i^b$	Jacobian relating to the floating-base joint for the leg $i$ CoM motion
$\mathbf{J}_i^a$	Jacobian relating to the leg $i$ actuated joints for the leg $i$ CoM motion
$\mathbf{J}_i^p$	Jacobian relating to the leg $i$ joints for the leg $i$ CoM motion
$\mathbf{J}_{c_i}^a$	Jacobian relating to the floating-base joint for the leg $i$ foot motion
$\mathbf{p}_{Gb}$	Relative position between the robot CoM and the base
$T_{sw_i}$	Desired step period
$\mathbf{S}_{xy}$	Selection matrix for the head two components of one 3D vector
$\mathbf{S}_x, \mathbf{S}_y, \mathbf{S}_z$	Selection matrix for the first, second, third component of one 3D vector
$\mathbf{S}_{control}$	Vector relating to the control signal for switching the swing-leg indexes
$\mathbf{S}_{contact}$	Vector relating to the contact states of four-leg feet
$n_{level}$	Control hierarchy number
$\mathbf{J}_{sw}$	Jacobian relating the motions of the swing legs
$\tilde{\mathbf{P}}_i$	Beginning position of leg $i$ CoM position at the robot normal state
$\mathbf{W}$	Mapping matrix enabling the joint torque to be consistent with all kinematics constraints





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivations and Objectives . . . . .	1
1.2	State of the Art . . . . .	3
1.2.1	State of the Art in Quadruped Robotics . . . . .	3
1.2.2	State of the Art in Legged-on-Wheel Robotics . . . . .	5
1.2.3	Legged Motion Generation . . . . .	7
1.2.4	Wheeled Motion Generation . . . . .	10
1.2.5	Inverse Dynamics & Operational Space Control . . . . .	11
1.2.6	Centroidal Dynamics . . . . .	13
1.2.7	Optimization based Torque Control . . . . .	14
1.2.8	Multi-Task Active Compliance . . . . .	16
1.3	Approach and Contribution . . . . .	17
1.4	Thesis Outline . . . . .	19
<b>2</b>	<b>System Modeling</b>	<b>21</b>
2.1	Floating-base Multi-Body Dynamics of Towr <i>ISIR</i> . . . . .	21
2.1.1	Mechanism Structure & Spatial Notations . . . . .	22
2.1.2	Wheel Contact Point Estimation . . . . .	26
2.1.3	Whole-body Inverse kinematics of Towr <i>ISIR</i> . . . . .	26
2.1.4	Generalized Dynamics Model . . . . .	28
2.1.5	Floating-base Centroidal Dynamics . . . . .	29
2.2	Operational Space Torque Control . . . . .	31
2.2.1	General Control Framework . . . . .	31
2.2.2	Operational Space Control (OSC) . . . . .	31
2.2.3	Contact-constraint Consistent Projections . . . . .	33
2.2.4	Contact Constraint consistent Prioritized OSC . . . . .	34
2.3	Summary . . . . .	35
<b>3</b>	<b>Prioritized Impedance Force based Hierarchical Torque Control</b>	<b>36</b>
3.1	Introduction and Chapter Outline . . . . .	37
3.2	Generalized Dynamics Model for a Locomotion Robot . . . . .	37
3.3	Whole-body Impedance Controllers . . . . .	38
3.4	Prioritized Impedance Controller based Torque Control . . . . .	40
3.4.1	Modified Dynamics Model for Pseudo-inverse based Dynamics . . . . .	40
3.4.2	Prioritized Impedance Controller based Null-space based Prioritized Weighted Optimization (P-WQP) . . . . .	41
3.4.3	One New Dynamics Model for Prioritized Impedance Controller based Hierarchical Optimization (P-HQP) . . . . .	42
3.4.4	Integration of the Prioritized Weighted Scheme in the Hierarchical Optimization (P-HWQP) . . . . .	47
3.5	Simulation . . . . .	48
3.5.1	Simulations for the Prioritized Weighted Optimization . . . . .	50
3.5.2	Simulations for the Hybrid Hierarchical-weighted Optimization . . . . .	52
3.6	Summary . . . . .	55

<b>4</b>	<b>Model-based Wheeled Locomotion</b>	<b>56</b>
4.1	Introduction and Chapter Outline	56
4.2	Wheel Motion Generator	57
4.2.1	Wheel Velocity Generator	59
4.2.2	Wheel Acceleration Generator	60
4.3	Whole-body Motion Generator	61
4.3.1	Adaptive altitude Control Model	62
4.3.2	Wheel-center Motion Relative to the Base	63
4.3.3	Whole-body Velocity Generator	64
4.3.4	Whole-body Acceleration Generator	64
4.4	Whole-body Torque Controller	65
4.4.1	Feedback Whole-body Impedance Controllers	66
4.4.2	PIC based Torque Controller	67
4.5	Simulation on Various Rough Terrains	68
4.5.1	Running on Artificial Environment	68
4.5.2	Running on 3D Rough Terrain	70
4.5.3	Running on Terrain with Big Altitude Difference	71
4.5.4	Running on Non-symmetric Geometry	72
4.6	Summary	73
<b>5</b>	<b>Model-based Legged Locomotion</b>	<b>74</b>
5.1	Introduction and Chapter Outline	75
5.2	Dynamics Modeling	76
5.2.1	Whole-Robot Dynamics Modeling	76
5.2.2	Each Leg Centroidal Motion	81
5.2.3	Whole-body Centroidal Jacobian	83
5.2.4	Centroidal Model Decompositions	84
5.3	Legged Motion Generation	85
5.3.1	New Quadruped Locomotion Principle	85
5.3.2	New Legged Motion Generator	86
5.3.3	External Motion Reference	87
5.3.4	Internal Motion Reference	89
5.3.5	State Machine	92
5.3.6	Modified Centroidal Models For Swing-leg Motion Generators	93
5.3.7	Desired Composite Swing-leg Motion	94
5.3.8	Desired Swing-leg Motion for Walking	95
5.3.9	Leg-Motion Relationship for Trotting	96
5.3.10	Desired Swing-leg Motions for Trotting	97
5.4	Whole-body Torque Controller	98
5.4.1	Prioritized Compliant Controller	99
5.4.2	Optimization based Torque Controller	101
5.4.3	Lateral Push-recovery Reaction	103
5.5	Simulations for Quad <i>ISIR</i>	104
5.5.1	Trotting Forward	105
5.5.2	Trotting Lateral, Turning in Place and Push-recovery	108
5.5.3	Trotting on Rough Terrain	109
5.6	Summary	111
<b>6</b>	<b>Conclusion and Future Works</b>	<b>112</b>
6.1	Conclusions	112
6.2	Perspectives	113
<b>A</b>	<b>Robot Structure</b>	<b>115</b>

<b>Contents</b>	<b>xi</b>
<b>B Simulation Environment</b>	<b>118</b>
<b>Curriculum Vitae</b>	<b>119</b>
<b>Bibliography</b>	<b>120</b>



# List of Figures

1.1	Legged Robotics in Boston Dynamics & MIT . . . . .	4
1.2	quadruped Robots in DLS lab of IIT . . . . .	4
1.3	Legged Robotics developed by ETH Zürich Robotic Systems Lab . . . . .	5
1.4	(a) Handle robot in Boston Dynamics, (b) Justin in German Aerospace Center (DLR) . . . . .	6
1.5	High mobility hybrid legged-wheeled robots . . . . .	7
1.6	Recent high mobility hybrid legged-on-wheel robots . . . . .	7
1.7	Spring Loaded Inverted Pendulum [Yu 2012] . . . . .	8
1.8	Central Pattern Generator [Barasuol 2011] . . . . .	9
1.9	Hylos Kinematic Model [Grand 2010] . . . . .	11
1.10	Legged-robot configurations with floating bases. . . . .	12
1.11	Centroidal Dynamics [Orin 2008] . . . . .	14
1.12	Cascade Impedance Controller [Semini 2016] . . . . .	16
2.1	“Towr <i>ISIR</i> ”—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 [Semini 2017] which has different leg layout and no wheels. Each joint is integrated with the force/torque sensor and the absolute/incremental encoder. . . . .	22
2.2	Structure of Towr <i>ISIR</i> 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. . . . .	23
2.3	Connectivity of Towr <i>ISIR</i> with body and joint indexes. . . . .	24
2.4	Whole-Body Dynamics Control Scheme of Towr <i>ISIR</i> . . . . .	32
3.1	[Left]: Task hierarchy definition. The smaller index, the higher priority. Index 0 denotes the contact constraint. For each priority, one specific impedance controller is applied. Other indirect impedance controllers can be integrated to be consistent with the task hierarchy using our approach in this chapter. [Right]: Towr <i>ISIR</i> is equipped with one manipulator on its floating base to enable the robot with manipulation behaviors. The manipulator has 6 DOFs and its mass is about 7kg. The weight of the robot with the manipulator is about 107kg. . . . .	38
3.2	[Left:] Connectivity of Towr <i>ISIR</i> : The robot has $N_B = 23$ bodies, the manipulator with purple color accounts for 6 bodies and the quadruped part has 17 bodies. The quadruped-CoM frame $G$ is set parallel with the base frame 1 or $b$ . [Right:] Indexes from 1 to 17 represent the Quadruped Bodies. Dashed circles represent the manipulator with body/joint index from 18 to 23. . . . .	39
3.3	P-HQP control framework: Given the quadruped-CoM translational motion reference, the legged and wheeled motions are achieved, detailed in Chapter 4. The motion references serve as the inputs of our dynamics controller in the dashed red box, including the proposed modified dynamics model, task extractions, prioritized impedance controllers, recursively updated inequality constraints and convex optimization. Three hierarchies are defined and each color stands for one hierarchy. The variables in this framework are explained in Chapter 2 and in this chapter. . . . .	43

3.4	Two simulation scenarios under GAZEBO simulator. (a) The robot runs along with pushing a door open, including the extension and retraction of the manipulator. (b) The robot runs on the rough terrain along with washing a wall, including the suspension of the legs and the manipulator motion. . . . .	50
3.5	Plots of robot states for two simulation scenarios with index 1 and 2 respectively. (1a) Plot of quadruped CoM motion in forward $x$ direction. (2a) Plot of quadruped CoM motion in forward $x$ and vertical $z$ direction. (1b) and (2b) Plot of the four wheels' rotational position errors: F, L, R, H represent front, left, right and hind. (1c) and (2c) Plot of the motion of the manipulator end effector: desired and actual orientations (roll, pitch, yaw) and positions ( $x, y, z$ ) of the end effector are relative to the base frame. . . . .	51
3.6	The robot is driven on various rough terrains while handling some activities. The whole process is divided into four stages. Stage 1: the beginning state. Stage 2: the robot holds the handrail using its arm. Stage 3: the robot retracts its arm after the handrail. Stage 4: the robot squats down and crosses through the desk and over one bump under the desk. . . . .	52
3.7	Plots of robot states by using the hybrid hierarchical-weighted control framework. The left axes denote the motion errors and the right axes represent positions. (a) plots the centroidal translational motion in the forward direction. (b) plots the centroidal translational motion in the vertical direction. (c) plots the front-left wheel rotational positions and the motion error. (d) plots the manipulator end effector motion errors, including the orientation errors (roll, pitch, yaw) and the position errors ( $x, y, z$ ). . . . .	53
3.8	CoM tracking performance comparison by the hierarchical scheme and the null-space based weighted scheme. (a) plots the centroidal position errors in the forward direction. (b) plots the centroidal position errors in the lateral direction. . . . .	54
3.9	Hierarchical control between the centroidal motion tracking and the wheel motion tracking tasks. (a) plots the centroidal position errors in the forward direction. (b) plots the wheel-joint motion errors. . . . .	55
4.1	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 4.2. 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 4.4. . . . .	58
4.2	Whole-body Motion Generation Framework: For the rolling locomotion mode, by given the quadruped-CoM translational motion reference. The superscript $ref$ means the reference. The whole-body velocity and acceleration are derived using the velocity and acceleration based models in the dashed blue and red boxes, detailed in Section 4.3. The outputs of the generators ( $\Upsilon, \dot{\Upsilon}$ ), the desired whole-body motions, serve as the inputs of the prioritized impedance controller based torque controller in the yellow dashed box, and $G$ and $w$ denote the centroidal motion and the wheel motion tasks. . . . .	62
4.3	The motion references are shown in the robot sagittal plane. $\mathbf{p}_{G[x]}^{ref}$ denotes the centroidal translational motion in the forward direction of the inertial frame. $\tilde{\mathbf{p}}_{G[z]}^{ref}$ represents the centroidal translational motion relative to the base frame in $z_b$ direction. $\tilde{\mathbf{p}}_{w_i[x]}^{ref}$ is the leg $i$ wheel-center motion reference relative to the base frame in $x_b$ direction. . . . .	63
4.4	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. . . . .	67

4.5	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. . . . .	68
4.6	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. . . .	69
4.7	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.	69
4.8	The robot runs forward on various rough terrains and handles several activities. The symmetric rough terrain geometry of the platform is depicted on the picture. The whole running process is divided into eight stages. We also show some zoom-in pictures of several stages. . . . .	70
4.9	Motion tracking results of the centroidal motion. The left and right axes of (a) and (c) represent the errors and positions respectively. (a) records the robot centroidal forward position and error. (b) plots the desired and actual CoM position relative to the base in the base vertical direction. (b) also plots the actual CoM height position in the inertial frame. (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 four wheel contact normal forces. . . . .	71
4.10	The robot runs forward on the unsymmetrical terrain, including (1) the slope in both forward and lateral directions, and (2) one platform with the half width of the road. The upper zoom-in pictures show the robot behavior on the slope indexed with (1). The left edge of the slope is $0.3m$ higher than the right edge. The bottom zoom-in pictures show that only the right two legs are the platform with index (2). The height of the platform (2) is $0.2m$ . . . . .	72
4.11	Two-task tracking performance relating to the centroidal motion and the wheel motion. The three figures have the same descriptions as in the caption of Fig. 4.9. . . . .	73
5.1	"QuadISIR"-A quadruped robot developed in ISIR lab. Each leg has three active-actuated joints, including the hip abduction/adduction joint (HAA), hip flexion/extension joint (HFE), knee flexion/extension joint (KFE). The fourth ankle joint is actuated by one mechanism structure. . . . .	75
5.2	Robot Configuration: Each leg is consisted with three actuated joints, HAA, HFE and KFE, whose frame $z$ axes are consistent with the joint axes. F, L, R, H represent front, left, right, hind. . . . .	77
5.3	New design principle of leg $i$ . In the sagittal plane, the hip point, leg CoM and the contact point are on the same line. . . . .	79
5.4	An example in the sagittal plane to show our new locomotion principle using two legs for tracking the robot centroidal motion. $\mathbf{v}_f$ and $\mathbf{v}_h$ denote the velocity of the front and hind leg CoMs respectively. $x$ and $z$ represent the front and vertical directions respectively. . .	86
5.5	Whole-body Control Framework: The motion references include the external and internal motion levels. The internal motion references are used for the motion generation of walking and trotting gaits. In the motion generator, we achieve the swing-leg motion using the CoM mass distribution model, the centroidal momentum/dynamics models and the models relating to legged motion relationships. The two level motion references and the outputs of the motion generators are used as inputs of the torque controller. The variables in the framework are explained throughout this chapter. . . . .	87



5.6	An example of the base forward motion reference with three Bézier curves, two curves for acceleration and deceleration, one linear curve for uniform velocity motion. The whole reference is divided into several equal phases. Each phase represents one swing-step period.	88
5.7	The motion relationship between the centroidal motion and the base motion on one slope. The dash line connecting the frame origins of the CoM and the base is set to be parallel with the $z_0$ or $z_G$ axis.	89
5.8	This shows two adjacent strides, consisting two swing-leg steps. In the first step period, the two swing legs do not land by following the $p_{Gb_z}$ reference, another reference (with red color) is added to enable the robot to adapt to the real-time states. The second curve recovers to the normal state on the flat terrain.	91
5.9	In the first step period, the step period is less than the defined value.	91
5.10	This shows the leg motion relationship for the trotting mode in the forward-backward direction as shown in (a1) and (a2), in the lateral direction as depicted in (b1) and (b2), and in the turning-in-place situations as presented in (c1) and (c2). We take the front-left (FL) and hind-right (HR) legs as the swing legs. The front-right (FR) and hind-left (HL) legs serve as the stance legs. The blue and red arrows denote the motion direction of the swing legs and stance legs respectively relative to the base. The green arrows represent the robot motion direction.	95
5.11	Snapshots of the forward trotting for $15.2m$ with $0.8m/s$ velocity. The robot squats down from the base height $0.53m$ to $0.5m$ , then accelerates from zero speed to $0.8m/s$ within $1s$ . Then the robot trots with constant speed $0.8m/s$ for $18s$ , then decelerates to zero speed using the final $1s$ .	103
5.12	Four leg-CoM motions in the sagittal plane when the robot trotting forward for 80 steps. This figure includes the acceleration and constant-speed periods. The curves inside the stable loops represent the transition period when the robot accelerates.	103
5.13	Four leg-foot motions in the sagittal plane when the robot trotting forward for 80 steps, which corresponds to Fig. 5.12.	104
5.14	Transition of the legged motion in $x_b$ axis when the robot accelerates and decelerates in the forward direction. After $t = 23s$ , the robot trots in place in the final target position.	105
5.15	Forward-trotting step period record for 80 steps.	105
5.16	[Up]: The periodic cycles by one filter. [Down]: Actual base pitch angle.	106
5.17	The generated torque for four-leg joints [N/m].	106
5.18	The detected contact states of four legs when trotting forward during $t \in [2, 10]$ .	106
5.19	Base and robot-CoM tracking performance for forward trotting motion.	107
5.20	Snapshots of the lateral trotting with $0.25m/s$ .	108
5.21	Base tracking performance for the lateral trotting.	108
5.22	Snapshots of the turn trotting in place with $\frac{\pi}{8}rad/s$ using 64 steps. The robot trot to turn in the counterclockwise and clockwise directions.	109
5.23	Base angular-motion tracking performance for turn-trotting motion.	109
5.24	[Left]: Snapshots of the push recovery by trotting in place with $0.5m/s$ as far as $6.5m$ . [Right]: Push recovery when the robot trots forward with $0.5m/s$ . The small figure inside denotes the virtual external force profile.	109
5.25	Snapshots of trotting on rough terrain with $0.5m/s$ using 210 steps.	110
5.26	Base spatial motion tracking performance for the trotting mode on rough terrain. [Up]: base translational motion tracking error. [Middle]: base angular motion tracking error. [Down]: selected desired and actual motion comparison along the whole process.	110
A.1	Whole-body structure of TowrISIR.	115
A.2	Hind half structure of TowrISIR shows the hydraulic control components.	116
A.3	Floating base layout and dimension of TowrISIR.	116
A.4	Electric Architecture of TowrISIR.	116
A.5	6-DOF manipulator structure for TowrISIR.	117

---

A.6	One leg structure of Tower <i>ISIR</i> . . . . .	117
A.7	One leg structure of Quad <i>ISIR</i> with parallelogram mechanism. . . . .	117
B.1	The components required by the simulation environment. . . . .	118



# List of Tables

1.1	A comparison of Hydraulic Quadruped Robots . . . . .	5
1.2	A comparison of Legged-on-Wheel Robots . . . . .	6
2.1	Parent and children relationships of one body in <i>TowrISIR</i> . . . . .	24
4.1	Algorithms for the Wheel Motion Generation . . . . .	60
5.1	Parameters design of one leg: $l_1 = 121.5mm$ , $l_2 = 250mm$ , $l_3 = 200mm$ , $l_4 = 250mm$ in Fig. 5.3. . . . .	80
5.2	Control Point Definition . . . . .	90
5.3	Task Control Hierarchy . . . . .	101



# Introduction

---

## Contents

<b>1.1</b>	<b>Motivations and Objectives</b>	<b>1</b>
<b>1.2</b>	<b>State of the Art</b>	<b>3</b>
1.2.1	State of the Art in Quadruped Robotics	3
1.2.2	State of the Art in Legged-on-Wheel Robotics	5
1.2.3	Legged Motion Generation	7
1.2.4	Wheeled Motion Generation	10
1.2.5	Inverse Dynamics & Operational Space Control	11
1.2.6	Centroidal Dynamics	13
1.2.7	Optimization based Torque Control	14
1.2.8	Multi-Task Active Compliance	16
<b>1.3</b>	<b>Approach and Contribution</b>	<b>17</b>
<b>1.4</b>	<b>Thesis Outline</b>	<b>19</b>

---

Mobile robots have fascinated the world for a long time as they show the high potential to be assistants and are helpful to humans in everyday life and to work in unstructured environment for dangerous and tedious tasks. Wheeled robots show great performance on flat geometries with high speed and less energy consumption, but they are easily stuck on rough terrains. In contrast, legged robots are not tied to discontinuous natural environments like mountain lanes or artificial paths like stairs, and they can move with agility to cope with disturbances or uncertainties. Many large projects in the world are researching these two kinds of robots to further close the performance gap between mobile robots in laboratories and in the natural environment, and serve in humankind society, industries and natural disasters. From a practical point of view, the quadruped-on-wheel robots would have more versatile locomotion modes by combining the wheeled motion and dynamic legged motion together. For mobile robots, especially for legged robots and legged-on-wheel robots, four coarse specifications: *versatility*, *speed*, *efficiency* and *robustness*, play important roles in dynamic performance interacting with the environment [Hutter 2013a]. However, the majority of today's existed robots are still quite far from this goal. In this thesis, we will initially research the system modeling, motion generation and torque control for the rolling-mode locomotion of a quadruped-on-wheel robot with active legged suspension. We also research the dynamic legged motion of a quadruped robot, which poses fundamental challenges requiring highly dynamics, versatility and compliance.

## 1.1 Motivations and Objectives

Wheeled vehicle is one of the most successful inventions in humans' history. It gave humankind a much faster and more convenient way to travel to destinations within time period decreased by several orders of magnitude. Few centuries later, researchers developed autonomous rovers and make them possible to explore the universe space, and execute missions in industries, agriculture, mining and defense, or may even serve as assistants at home. They provide great potential applications using high speeds, consuming low energy with high efficiency on regular, flat, slopped or continuous surfaces. However, wheeled rovers reach their limits when handling very rough terrains. Approximately half of the terrestrial land and living

space on earth are inaccessible and unavailable to wheeled vehicles while natural animals and humans traverse such terrains with much ease and elegance. Therefore, the use of legged walking machines, especially legged robots enable locomotion possibility on challenging and changing environments, even without a continuous terrain road. Along the natural evolution, legs provide animals with amazing gait motion. Mountain goats have excellent rock climbing and balancing skills, horses and tigers can run quite fast with elegant gaits and through the forest with agility respectively, humans select different footholds to cross over obstacles with the size of our leg length. To overcome different surfaces, animals and humans unite various gaits with different speeds which are energetically efficient, and apply active suspension to interact with environment compliantly. Research in legged robots provides mankind possibility to build systems to assist human beings in their daily life, which can run on rough terrains and cross over rough terrain with discontinuities using highly dynamic locomotion gaits. Legged robots have the potential and numerous promising applications in dangerous areas like disaster rescue, volcano and forest exploration, as well as in construction sites, and legged robots even show superiority in the places that wheeled robots work on. However, legged robots run much slower and consume higher power than wheeled robots. In recent years, some of the legged robots are developed with versatility, speed, efficiency and robustness to some extent. However, there still remains a big gap when comparing with humans and animals, especially the developed approaches are far from the quadruped-animal locomotion principle resulting in each one of locomotion abilities requiring one specific method. Especially in the state of the art, the quadruped legs are always assumed to be massless which is unreasonable. The light legs experience much bigger acceleration than the floating base. Therefore, our motivation in this thesis is to seek a new locomotion principle for quadruped robots in this thesis. We believe that centroidal properties will enable the robot to generate the legged motion automatically on-line.

From the practical point of view, autonomous robotic vehicles and legged robots have their own strengths and limits, which gives us inspiration that they can be combined together to be a hybrid legged-on-wheel robot with both strengths. This kind of robots can adapt with rough terrains with active legged suspension, and can cross over artificial architectures. By combination of the dynamic legged and wheeled motions, the robot can achieve more types of locomotion, like rolling, walking, trotting, and even jumping. Compared with wheeled rovers and legged robots, this combination of both faces great difficulties in system modeling, motion generation and whole-body torque control. In the state of the art, most of the works assume that the legs are massless and do not take the advantage of the robot centroidal properties. In this thesis, we control the quadruped-on-wheel robot to track the centroidal motion reference and enable the robot to adapt to terrains with compatible behaviors. For example, the robot could run on terrains with much-difference altitude without specific control of the floating-base height.

In this thesis, we design two robots, including a quadruped-on-wheel robot named *TowrISIR*, and a quadruped robot, *QuadISIR*, which share the similar structure of the floating bases. Centroidal dynamics is one specific framework by controlling the robot centroidal properties to achieve a vast range of natural emergent whole-body behaviors. We integrate the centroidal momentum and dynamics models in the motion generators of the two robots to grant our robots with elegant and natural locomotion abilities. The outputs of the motion generators include the whole-body desired configuration, velocity and acceleration. Then we will use several improved whole-body torque controllers to verify these motion generators.

Although more sophisticated, the whole-body torque control enables the robot to have more natural behaviors and easier adaptations to external influences in comparison with kinematics position control. The more dynamic and versatility guarantee more robustness and efficiency of mobile robots while interacting with the changing environment. The operational-space control (OSC) is a more specific framework for controlling tasks' motion for any point on the robot. Due to high degrees of freedom (DOF), legged robots can maneuver multiple tasks at the same time by coordinating them in a reasonable way, by applying the null-space based orthogonal projections, and quadratic programming based optimization approaches to handle inequality constraints. Highly dynamics enables mobile robots to handle the interaction forces and obtain abundant desired dynamic behaviors with compliance. In this thesis, we will develop a general dynamics model which can embody the multi-level hierarchy framework and maintain a constant number of constraints. The new dynamics model will be used to synthesize several

optimization based torque controllers, then we will solve how to integrate and prioritize the general impedance controllers.

So far, the system design, motion generation and torque control are rarely synthesized and integrated in one controller, especially for quadruped-robot locomotion, which limits the achievement of highly dynamics and versatility in a natural and elegant way. Famous robots in Boston dynamics fascinate the world with videos that demonstrated their abilities to perform highly dynamic tasks (e.g. running or push recovery), but no details have been published. This strongly motivates us to address these topics, especially to integrate them in a coordinated way to provide the legged robots with highly dynamics, compliance, versatility and torque controllability. The motion generation and torque control methods explored in this thesis can also be used in other four-limb robots which can reduce the gap with the natural counterparts.

## 1.2 State of the Art

In this section, we will overview relevant research about quadruped robots and legged-on-wheel robots with concentrate on recent achievements and research methods that have an impact on our research. In the first two Sections 1.2.1 - 1.2.2, quadruped robots and legged-on-wheel robots are reviewed on design, actuation principles and applications. Then the approaches on motion generation and motion control are presented in Sections 1.2.3, 1.2.4, 1.2.5, 1.2.6, 1.2.7, 1.2.8, relating to the spring loaded inverted pendulum, central pattern generation, floating-base inverse kinematics and dynamics, centroidal dynamics, operational space control, active compliance, and hierarchical quadratic programming.

### 1.2.1 State of the Art in Quadruped Robotics

The category of legged robots relating to quadrupeds fascinate robotists due to their friendly outlook and mammal structure with four limbs, which widely broadens the scope of possible applications. The structure of quadruped robots allows them to interact with environment using more types of gaits, and with human in a more safe manner. Fig.1.1 shows the most advanced quadruped robots in the world. LS3 and BigDog [Buehler 2005, Raibert 2008] are manufactured to be support systems which perform fascinating balance skills after being kicked on ice surface, and their actuators are driven by hydraulic systems powered by engines. Spot is a light quadruped robot, powered by electric battery, it can handle highly dynamic behaviors like trotting and running. WildCat is the world's fastest quadruped robot, and can gallop like a horse while maintaining its balance. MIT Cheetah II is a recent example that performs tasks like running and jumping, thanks to its high power source and low geared electromagnetic motors with proprioceptive force control [Wensing 2017a]. MIT Cheetah II and BigDog are equipped with four degrees of freedom (DOF) for each leg while most of other quadruped robots have three DOFs for each leg. Spot, Spotmini and WildCat use M-configuration legs, LS3 with O-configuration and the other quadruped robots apply X configuration, although there is no proof about which configuration is better. Except that BigDog use physical springs as passive compliance under its feet, the other quadruped robots in Boston Dynamics and MIT select to use active compliance to interact with frequently changing environments. However, little is published about the design details and control methods of the quadrupeds in Boston Dynamics.

Compared with Boston Dynamics, very promising research results are published along the development of hydraulic HyQ, HyQmini and HyQ2Max in IIT's Dynamic Legged System lab in terms of weight, size, and actuation principle, shown in Figure.1.2. The main contribution of HyQ series is the novel design and research of torque control for agile and versatile quadruped robots with hydraulic actuation powered by on-board battery or off-board electricity. These three robot all have legs with X configuration, and their differences lie on their robot size, actuator types (vane rotary actuator, linear cylinder) and transition mechanisms at knee joints (such as the hinge joint, four-bar linkage, crossed four-bar linkage). A comparison of these advanced hydraulic quadruped robots has been made in Tab. 1.1, in which JINPOONG[Kim 2013] and RLA-1[KAWABATA 2014] are not torque controlled. Detailed overview of design and actuation principles can be found in [Semini 2010], [Khan 2015], [Semini 2017],



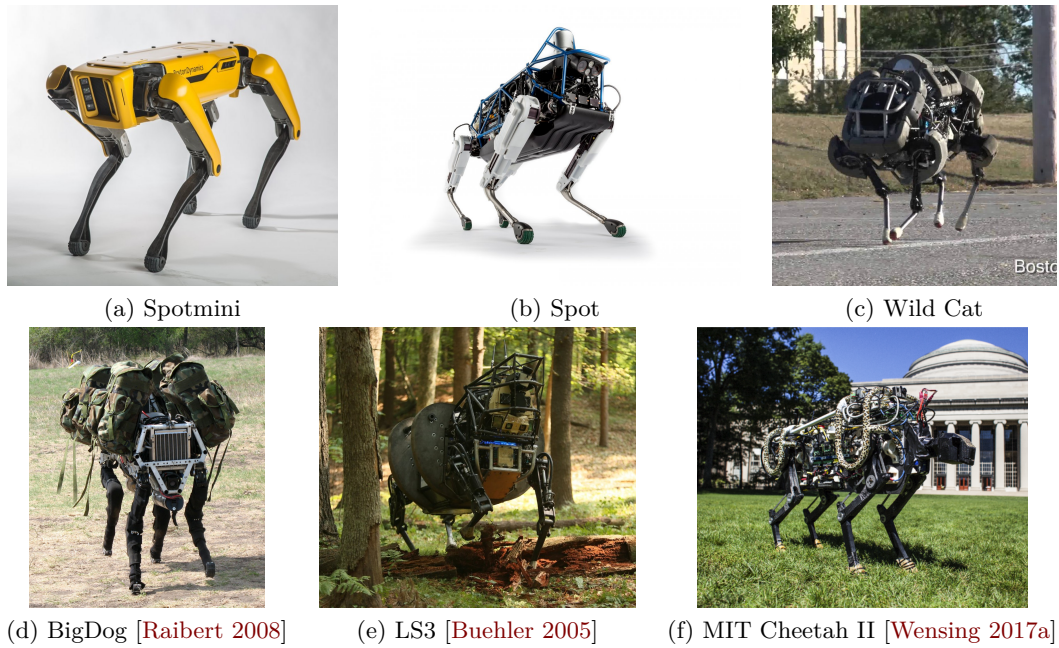


Figure 1.1: Legged Robotics in Boston Dynamics & MIT

especially about the novel design of quadruped robots capable of performing trotting, crawling, running over flat/uneven terrain and balancing, and about detailed selection methods of suitable hydraulic cylinders, valves, accumulators, pumps properties and linkage parameters.

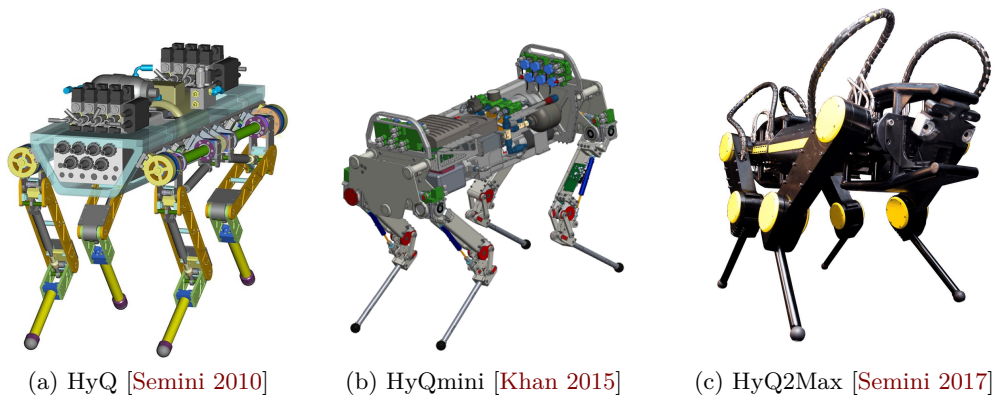


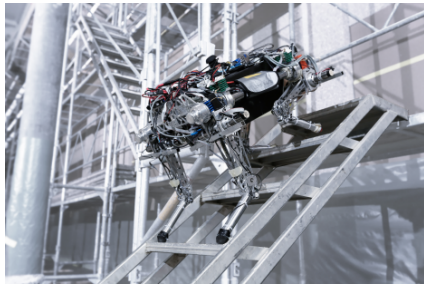
Figure 1.2: quadruped Robots in DLS lab of IIT

Classified by compliance types, quadruped robots in Fig.1.1 and Fig.1.2 use active compliance with flexibility to change the dynamic characteristics (such as stiffness and damping) through software. The closed-loop control of a hydraulic servo valve (MOOG E024: bandwidth>250Hz) and the hydraulic cylinder (Hoerbiger LB6) has the bandwidth of more than 100Hz. Whereas for the physical compliance, the bandwidth of the model-based control of robots in Fig.1.3 is 11Hz for StarLETH [Hutter 2011] and 24Hz for ANYmal [Hutter 2016]. These two robots are developed in ETH Zürich Robotic Systems Lab and showed locomotion capabilities of walking and trotting. The joint modules of StarLETH are designed with three kinds of SEA (Series-Elastic Actuators) arrangements with different connection order of the motor, gear and two linear symmetrical compressed springs, while ANYmal integrates high-torque actuators, and harmonic drive gears along with rotational springs in one compact and compliant joint unit. Although the mechanical compliance in SEA-controlled robots functions as protection of impacts

Table 1.1: A comparison of Hydraulic Quadruped Robots

Robot Name	Country	Year	Mass (off/onboard power)	Dimension (LxWxH)	DOF (per leg)
JINPOONG	Korea	2013	N/A, 120kg	1.1m x 0.4m x 1.2m	4
RLA-1	Japan	2014	60.2kg, N/A	1.1m x 0.67m x 1m	3
BigDog	USA	2008	N/A, 109kg	1.1m x 0.4m x 1m	4
LS3	USA	2012	N/A, 590kg	NA x NA x 1.7m	3
WildCat	USA	2013	N/A, 154kg	NA x NA x 1.17m	3
Spot	USA	2015	NA, 75kg	NA x NA x 0.94m	3
HyQ	Italy	2011	75kg, 98kg	1m x 0.5m x 1m	3
miniHyQ	Italy	2015	24kg, 35kg	0.85m x 0.35m x 0.77m	3
HyQ2Max	Italy	2015	80kg, N/A	1.30m x 0.54m x 0.92m	3

from the external forces, it additionally limits the control bandwidth required by highly dynamics and compliance. Therefore, these kinds of joint modules need careful design with bandwidth compromise. In addition, the powerful electrical-actuator based Spotmini and MIT Cheetah II [Wensing 2017a] showed highly dynamic locomotion abilities without using passive compliant components.



(a) StarlETH



(b) ANYmal

Figure 1.3: Legged Robotics developed by ETH Zürich Robotic Systems Lab

### 1.2.2 State of the Art in Legged-on-Wheel Robotics

Legged robots fascinate the public which have the potential to replace humans in dangerous and dirty duties. They are expected to operate in unstructured outdoor areas, such as collapsed buildings, disaster sites, forests, mountains and universe space. Different applications require robots with more diverse locomotion modes, efficient and compliant behaviors. They should be capable of finishing tasks in limited time period. A promising solution is to mount legged robots on wheels to combine vehicles' efficiency and legs' rough-terrain capability, leading to legged-on-wheel robots. They can enjoy the vast diversity of locomotion performance of legged robots, and further can roll on continuous rough terrain with high speed and less energy consumption. Boston Dynamics proposed an agile high-strength mobile manipulator, named Handle shown in Fig.1.4(a), powered by hydraulic and electric actuators for ten joints and two wheels, it can rotate with zero radius, roll through stairs, jump up on a high desk with active compliance and natural whole-body balance. . However, no research paper on Handle has been made public. Justin in Fig.1.4(b) is designed by DLR with humanoid upper body mounted on a platform which connects four wheels with passive legs. Justin is designed to serve in human life as assistants and it focuses on abundant manipulation behaviors without the necessity of considering robot balance and stabilizing the legged motion which is essential for legged robots [Dietrich 2016]. The research results of Justin are mainly about the whole-body torque controller for humanoids using redundancy resolution and active compliance control for physical interactions [Dietrich 2015].



(a)



(b) [Dietrich 2016]

Figure 1.4: (a) Handle robot in Boston Dynamics, (b) Justin in German Aerospace Center (DLR)

Table 1.2: A comparison of Legged-on-Wheel Robots

Robot Name	Length	Weight	Speed	Obstacle Height
Nomad	2.4m	725kg	1.8km/h	N/A
Curiosity	3m	900kg	0.15km/h	50cm
Shrimp	0.6m	3.1kg	N/A	22cm
Athlete	6m	1200kg	10km/h	N/A
WorkPartner	1.4m	160kg	12km/h	N/A
Multi Tasking Robot	N/A	N/A	25km/h	43cm
Mirco Hydraulic Toolkit	0.7m	150kg	N/A	N/A
Hylos	0.6m	12kg	N/A	N/A
Hylos II	0.54m	20.5kg	1.8km/h	19cm
Complios Robot	0.58m	11kg	N/A	N/A

Other legged-on-wheel robots are the combination of multi-leg robots and wheeled rovers, amongst which Sojourner [Volpe 1996], Shrimp [Siegwart 2002], Nomad [Rollins 1998], Curiosity [Grotzinger 2012] are articulated multibody structures to permit passive adaptation to the ground. While rovers like SRR [Iagnemma 2003], Gofor [Sreenivasan 1994] and Lama [Andrade 1998] use active suspension to prolongate their abilities on rough terrain, they can only roll on continuous ground by maintaining contacts without legged locomotion skills, which are not suitable enough for complex missions in construction sites, rocky mountains or universe space with infinite geometry possibilities. High mobility hybrid systems such as ATHLETE lunar vehicle [Hauser 2008] and Tri-ATHLETE [Heverly 2010] developed by NASA/JPL with six legs serve for lunar exploration. Multi Tasking Robot (MTR) [Bouloubasis 2007] has four "wheel-paw" assemblies allowing it to cross obstacles with 2.5 times the diameter of its wheels. WorkPartner [Yleonen 2002] with four legs developed by Helsinki University of Technology is able to interact with humans with hybrid locomotion in daily life. MHT (Micro Hydraulic Toolkit) [Thomson 2012] is developed by Defence R&D Canada-Suffield, powered by hydraulics for unknown terrain exploration, which brings the similar control methods from Hylos robots that compute the inverse kinematics while tracking a desired floating-base posture and trajectory. These three robots are shown in Fig.1.5(a), (b) and (c). In addition, Tab. 1.2 lists characteristics of these robots. Although inheriting both advantages of wheeled and walking systems, they are controlled by whole-body kinematics and optimize posture to keep balance.

Very promising research results come along with the design and control developments of Hylos series [Grand 2010, Grand 2016] and Complios Robot [BOUTON 2017a] developed by Institut des Systèmes Intelligents et de Robotique (ISIR) of Sorbonne Université, as shown in Fig.1.5(d), (e) and (f). Hylos and Hylos II share similar mechanical structures: they are composed of four wheel-legs, and each wheel-leg consists of a multi-DOF series chain ended by an actuated and steerable wheel. They are capable of changing the position of its center of mass (CoM), optimizing the joint actuated torques using the robot structure redundancy and distributing the contact forces, which maximize the stability. Complios is



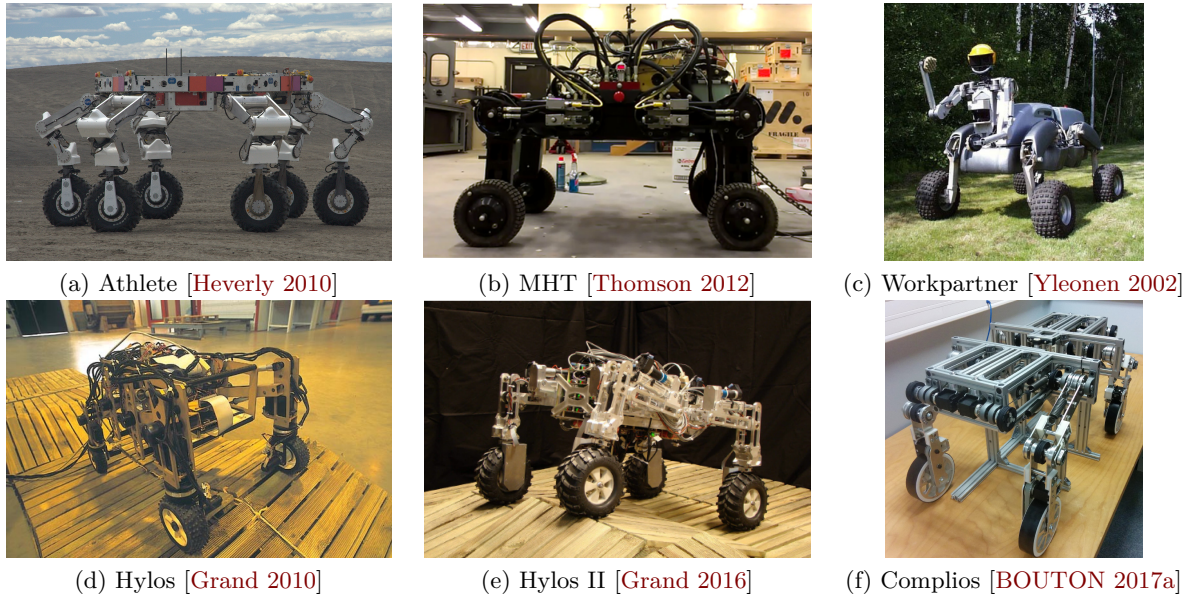


Figure 1.5: High mobility hybrid legged-wheeled robots

designed with active compliant legs driven by SEA which can react to obstacles by Q-learning approach while adapting posture to keep balance [Bouton 2016, Bouton 2017b]. Recently, one centaur-like robot with four wheel-legs, CENTAURO [Kamedula 2018][Kamedula 2020] shown in Fig. 1.6(a) was developed to track the system centroidal motion using the whole-body kinematics. However, this work only uses the feedback torque controller. Another recent quadruped-on-wheel robot is shown in Fig. 1.6(b) [Bjelonic 2020], and it is the combination of the quadruped robot ANYmal and electrical wheels, in which the base motion is tracked without applying the extra centroidal properties, detailed in this thesis.



Figure 1.6: Recent high mobility hybrid legged-on-wheel robots

### 1.2.3 Legged Motion Generation

Robot motion generation deals with the methods that can create behaviors of the system. Planning, optimization, and learning is one systematic method that are used by projects in DARPA Learning Locomotion Challenge [Kalakrishnan 2010]. However, the combination of a high precision ground modeling, extensive foothold search and motion planning is not suitable for reactive gait planning in unknown rough environment. In this section, two methods of motion generation are overviewed in terms of model accuracy, efficiency and spontaneous pattern. Inspired from nature, humans and quadruped animals choose gait patterns and footholds to propel their torso forward, react to external disturbances and cope with challenging terrain. They choose different and optimal gaits to travel at a certain speed or to ro-

bustly handle unanticipated disturbances, and they show perfect energy efficiency. Biomechanists have been focusing on the fascinating locomotion principles to imitate complex whole-body dynamics motion using simplified templates which can mimic the important locomotion features of legged robots. The Linear-Inverted Pendulum (LIP) template [Winter 1995] is the most commonly used model to explain human balance and walking which captures the relationship between the ZMP and the evolution of the CoM. Although LIP has been widely used by humanoid stably walking [Kajita 2001], it has limitations relating to highly dynamic behaviors such as running. Another widely used template is the Spring Loaded Inverted Pendulum (SLIP) [Blickhan 1989] which is also a simplified model, shown in Figure 1.7. It replaces the LIP rigid body by a massless spring as an appropriate low-order model to describe the CoM dynamics and ground reaction forces of human and animal gaits with astonishing accuracy in running across a wide range of sizes and morphologies [Blickhan 1993]. In contrast to the LIP model which represents stable behaviors, SLIP captures stance and flight phases that exist during trotting and galloping, and explains some dynamic self-stable properties of locomotion found in nature. For example, during each individual step of running, the system falls forward, but it is able to catch itself using the following foot step and repeat the pattern continuously. After the robot is disturbed in the forward and lateral directions, the robot can asymptotically recover back to its normal and periodic motion. The kinetic and potential energy interchange cyclically to yield controllable and efficient locomotion. SLIP template has been used to generate single-leg hopping and quadruped trotting in MIT Leg Lab in the 1980s [Raibert 1984, Raibert 1989] by intuitive observations [Raibert 1986] and further be applied on BigDog in Boston Dynamics.

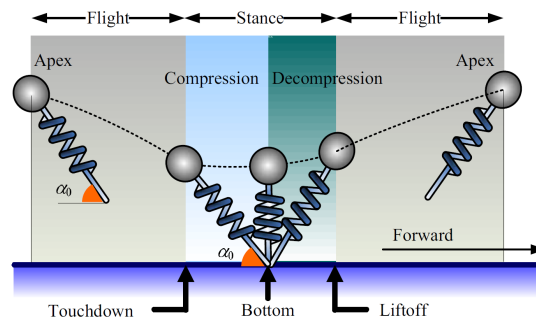


Figure 1.7: Spring Loaded Inverted Pendulum [Yu 2012]

Moreover, various control strategies like dead-beat controller are developed in addition to the passive dynamic self stabilization, to enable the disturbance rejection within short period by adjusting appropriately the leg spring stiffness or the touch-down angle [Seyfarth 2002]. It means that a constant velocity and apex height can be maintained and controllable even on extremely unknown terrain [Ernst 2010]. Other researchers seek accurate and analytic methods to approximate the nonlinear dynamics model which is non-integrable under the gravity effect during the stance phase. Some recent available methods are developed critically for stability analysis and approximation of the legged locomotion, based on some unrealistic assumptions, e.g. the conservation of angular momentum, and they neglect the damping loss [Schwind 2000, Geyer 2005]. [Arslan 2009] incorporates the total gravity effect on the angular momentum to achieve good tracking performance of non-symmetric trajectories. [Yu 2012, Han 2014] adjust spring stiffness in stance phases to compensate energy waste between the current and desired apex states. [Saranli 2010] further takes the presence of non-symmetric steps and passive damping under the gravity effect into consideration, ensuring the approximation methods to be more realistic. [Hutter 2010, Hutter 2014] projected the robot behavior of the passive SLIP model onto the dynamic motion of an actual robot leg by utilizing an operational space controller, then different types of dynamic-gait motions are generated [Gehring 2013]. [Wensing 2014b, Wensing 2013b, Wensing 2014c] use one 3D-SLIP model to control high-speed running, steering and jumping of a humanoid by adjusting its foot-touchdown locations and CoM trajectories combined with Prioritized Task-Space Control [Wensing 2013a]. [Geyer 2006] proposed another template called the Dual Spring-Loaded Inverted Pen-

dulum which is a less-studied template model for walking. Then [Liu 2015] extended it to a 3D version to achieve periodic high-speed walking gaits. [Yu 2017] unitizes two-layered Dual-SLIP to achieve stable galloping gait and applies the task-space formulation for a torque controlled quadruped robot with spine joints. A dead-beat controller based approximation mapping method [Yu 2012] provides desired apex state as well as two CoM-trajectory references, and one prioritized multi-task controller enforces the dual-CoMs to behave the target dynamics. However, when multi-SLIP models are applied for humanoids or quadrupeds, each leg CoM cannot imitate the real mass distribution of the whole robot. In addition, when SLIP is applied for slow motion, e.g. walking, the passive template is not suitable.

Another legged motion principle is based on the central pattern generator (CPG). Inspired by human and quadruped animals, CPG can emulate the neural circuits which are found in invertebrate and vertebrate animals that can produce rhythmic patterns of neural activities (such as chewing, breathing, digesting) without any rhythmic inputs from higher control centers or sensor-based feedback signals. For example, in the locomotion system of vertebrates, the spinal CPGs serve to produce the coordinated motion patterns, and higher control centers serve to modulate these rhythmic patterns concerning the environmental conditions [Ijspeert 2008]. Recently, CPGs served as mathematical models for trajectory generation of legged robots to produce stable rhythmic motion using limit cycle patterns, and require very few control parameters that enable the locomotion modulation by integrating sensory feedback signals [Wu 2009, Barasuol 2011]. Majority of CPGs applied for trajectory generation are established in joint space [Morimoto 2008, Buchli 2007]. This method needs to map the feet trajectories with simple signals into joint space that leads to complex non-intuitive signals in joint-space CPGs which are not easily modeled by few harmonics. Furthermore, there are also coupling effects in some cases among the parameter modulations for joint-space CPGs to obtain task-space trajectory references. [Liu 2011] proposed an operational-space CPG, however, its CPG parameters are still non-intuitive and need to be mapped to the gait parameters. To this end, [Barasuol 2013] contributed to this field with CPG-inspired task-space trajectory generator, as shown in Fig. 1.8, in which all the parameters have intuitive meanings such as step height, step length and velocities, and can be adjusted to influence each foot trajectory independently. Combined with coupling matrix for trotting, walking, bounding and pacing gaits and by using the exponential functions for smooth-trajectory transitions [Righetti 2008], this CPG-inspired trajectory generator adjusts its oscillator’s horizontal plane to coincide with current geometry and is kinematically compliant when traversing challenging terrain without any terrain maps. Despite of its good performance for task-space pattern generation, feet trajectories cannot be easily adjusted and the CPG method is not suitable at the onset of the starting motion, during the running-mode transition and when the desired trajectories are changing all the time to deal with the unknown environment.

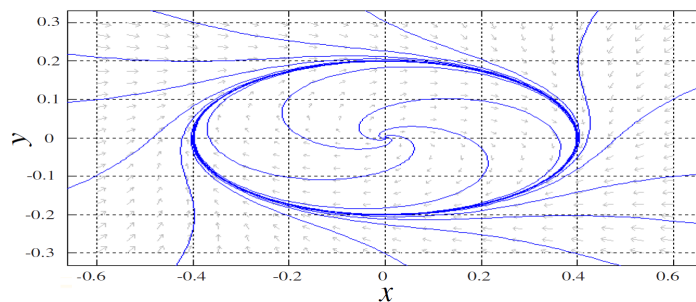


Figure 1.8: Central Pattern Generator [Barasuol 2011]

In addition, [Hyun 2014] proposed the usage of Bézier curves to generate 2D trajectory of each leg in which the step height can be controlled to do obstacle clearance, although intuitive, the motion generation is still in kinematics level. [Park 2017] and [Park 2015] apply the Bézier curves to scale the contact impulse for the stance phases in dynamics level to generate the bounding and jumping motions. However, the simplified model cannot account spatial motion, especially the spatial momentum conservation. These approaches generate the legged motion first, then the base motion is generated online which is not consistent with the quadruped-locomotion principle. Although these methods enable

fast calculations and achieve several quadruped gaits, the hand designs are required and impractical to satisfy too many kinds of behaviors, e.g. standing up and climbing stairs. Therefore, a general legged locomotion generator should be developed to generate various kinds of gaits automatically.

Due to the existence of internal and external dynamics constraints, model-based optimization methods have recently been developed for legged robots to generate optimal trajectories in the receding horizon. They rely on real-time trajectory optimization to predict the outcome of possible actions and find an optimal future plan. In recent years, model based trajectory optimization (TO) is applied to generate more abundant behaviors considering the changing environment and robot-self ability limits. A MPC-based CoM trajectory re-planning has been originally implemented for bipeds in [Wieber 2006] by using simplified ZMP-based dynamics model. [Aceituno-Cabezas 2017] and [Deits 2014] apply the Mixed-Integer convex optimization to determine gait transitions, contact locations and different types of motions which accounts for various constraints and is not limited to a pre-specified gait sequence. Authors in [Dai 2014] and [Posa 2014] applied the Sequential Quadratic Programming (SQP) method to resolve contact constraint forces and optimize the trajectory by satisfying complementary constraints. [Winkler 2018] uses the Interior Point method to generate different kinds of gaits and trajectories for legged robots through phase-based end-effector parameterization. However, most of the models for TO of quadruped robots are simplified by assuming massless legs which present much gap between the simplified model and the real robot. [Mastalli 2016] computes a feasible robot-CoM trajectory by mapping the contact forces and applying a hierarchical optimization framework to handle the variant dynamics model with different contact situations. [Mastalli 2020] proposed the FDDP (Feasibility-driven Differential Dynamic Programming) algorithm which can generate the whole-body trajectories of legged robots within 0.5s. Authors in [Melon 2020] propose to combine the learning and optimization methods in which the learned initializations speed up the convergence to the optimal trajectory in a 16-step horizon. However, for predictive TO problem, due to the fact that differentiation and integration are really difficult and complex, discretization of the dynamics model is always applied to states, control inputs, contact forces, etc, which makes the optimization problem really huge. Approximated integrations and differentiations require long calculation time which is not suitable for real-time application. Therefore, the optimization based motion generators are not suitable currently for real-time motion generation until CPUs can run more 30-100 times faster.

### 1.2.4 Wheeled Motion Generation

For legged-on-wheel robots, the promising results of trajectory and posture control are along the development of the series Hylos robots in ISIR shown in Fig. 1.5. The differential kinematics model (1.1) is initially derived to get the relationship between the articular-joint parameters and the platform parameters of all wheel-legged chains by assuming the pure rolling condition [Grand 2004], given by:

$$\mathbf{L}(\mathbf{x}, \Theta, \mathbf{n})\dot{\mathbf{x}} + \mathbf{J}(\Theta, \mathbf{n})\dot{\Theta} = \mathbf{0} \quad (1.1)$$

where  $\mathbf{L}$  is the matrix relating to the base movement  $\mathbf{x}^T = (\boldsymbol{\rho}, \boldsymbol{\phi})^T = (x, y, z, \alpha, \beta, \gamma)^T$ ,  $\mathbf{J}$  is the motion Jacobian of all wheel-legged differential kinematic chains.  $\Theta_i^T = (\boldsymbol{\theta}_i^T, \boldsymbol{\chi}_i^T)$  in which  $\boldsymbol{\theta}_i$  and  $\boldsymbol{\chi}_i$  denote the vectors of leg  $i$  joint parameters and wheel's parameters. Similar inverse kinematics methods can be found in wheeled rovers [Oriolo 2002], humanoid-on-rover robot [Giordano 2009], mobile manipulators [De Luca 2006, De Luca 2010].

[Grand 2004] used the kinematic model in (1.1) to control the generalized trajectory of the robot and treated trajectory and posture control together. A velocity based posture control algorithm calculates the joint velocities to reach desired posture by following the desired trajectory. This generic kinematic model is applied to find optimal posture of Hylos to control the CoP (the distribution of contact forces) and the tip-over stability margin that is proposed by [Papadopoulos 1996]. [Besseron 2008] modified the posture to ensure the robot stability without the necessity to specify a posture state by using one potential field [Khatib 1986] w.r.t. to stability threshold. [Grand 2010] included non-ideal contact conditions into the kinematic model in (1.1) and derived the force transmission between contact forces, joint torques and gravity force. In addition, [Grand 2016] extended the control methodology for step obstacle crossing,



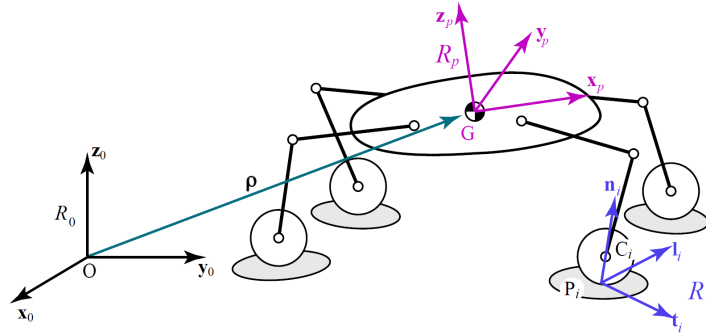


Figure 1.9: Hylos Kinematic Model [Grand 2010]

which optimizes the system posture and the distribution of internal forces while respecting the frictional contact constraint, by taking the advantages of the robot structure redundancy to maximize its stability. However, a whole-body dynamics model is not applied which hinders the robot to behave dynamically. The Complios robot [Bouton 2012] is developed with both passive and actuated joints, and the active ones are driven by SEAs to react to obstacles. However, the passive joints and SEAs' low control bandwidth limit the robot dynamic motion on unknown rough terrain. Another hydraulic legged-on-wheel robot is developed in [Thomson 2012] and it has the similar control method with the Hylos robot that computes the inverse kinematics to track the desired CoM trajectory and optimizes the posture redundantly. The Momaro robot in [Klamt 2017] shows a hybrid locomotion mode to overcome obstacles like artificial stairs using a kinematics approach. However, the robots in [Grand 2004], [Bouton 2012], [Thomson 2012] and [Klamt 2017] only show static maneuvers, and the kinematics controller or the simplified dynamics model limits the dynamic behaviors. The balance stability are also ensured by considering the robot center of mass (CoM) position, the zero moment point (ZMP) [Vukobratović 2004] and the center of pressure [Popovic 2005]. Authors in [An 2012] and [Suzumura 2013] used the ZMP criteria to generate the robot CoM translational motion pattern to follow the ZMP reference in 2D and 3D space respectively. Then [Suzumura 2013] further generated the whole-body motion by considering the centroidal angular momentum model. The work in [Suzumura 2013] is extended by authors in [Nagano 2015] to achieve the contact kinematics in the acceleration level. However, above works always assumed that the legs are massless. This assumption is not suitable for highly dynamic motions since the robot CoM should combine all body masses and inertias which constitute the dynamics model and influence the whole-body motion generation. Recently, the centaur-like robot CENTAURO [Kamedula 2018][Kamedula 2020] tracks its centroidal motion, and the kinematics for the wheel-legged motion 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 generalized dynamics model or the centroidal dynamics model. In addition, the derivative of the kinematics contact constraint cannot be ensured without a feed-forward torque controller [Sentis 2005]. Instead of applying the ZMP index with the contact-coplanar limits and other disadvantages stated in [Caron 2015], authors in [Hutter 2014][Orsolino 2018][Caron 2015] show that quadruped or humanoid robots can embody feasible motions and dynamic stability due to the satisfaction of the contact constraints in the kinematics first and second orders enable. The contact constraints and the unilateral limits can be ensured by applying an optimization based torque controller which can also handle the friction cone. This kind of torque controller can also be used for quadruped-on-wheel robots which is neglected in the state of the art.

### 1.2.5 Inverse Dynamics & Operational Space Control

The dynamics model of a complex system presents the relationship between the robot motion and the generalized forces that act on the robot, which considers parameters such as body lengths, masses and inertias [Sciavicco 2014]. For operational space tasks such as locomotion gaits and posture, in contrast



with the inverse kinematics control method, dynamics control enables much more elaborated control concepts and can achieve far more natural elegant behaviors, especially when robots are required to operate and interact with humans in a dexterous and safe way. In addition, inverse dynamics control enables more compliant interaction properties and can also ensure good tracking performance to desired motion references. Fig.1.10 illustrates the 6 virtual degrees of freedom attached from inertial frame to the floating base so that whole system is hence underactuated when there are fewer contact constraints than floating-base DOFs [Righetti 2011b].

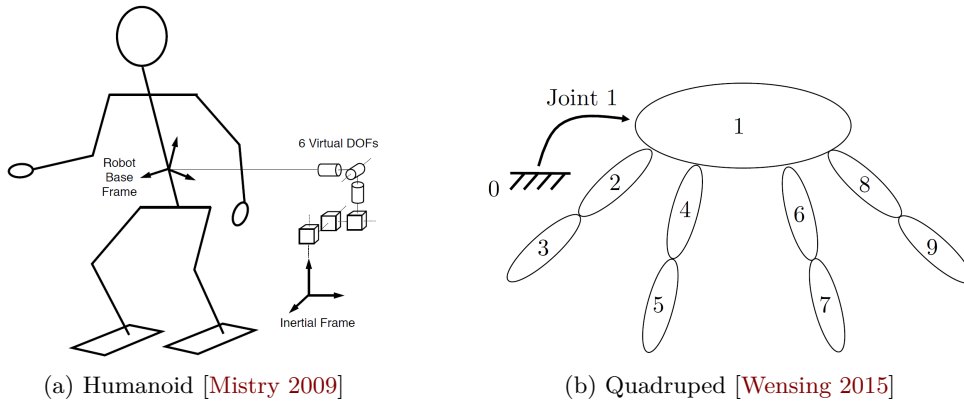


Figure 1.10: Legged-robot configurations with floating bases.

The solutions of the model based dynamics can be divided in the joint space and operational space. The former formulation is the classical approach which generates the necessary joint torques to achieve a desired movement in joint space that functions as a feed-forward command. In addition, an error based feedback command also in joint space, through active compliance controller with virtual low gains, can be added to handle movement derivation inducing from the model uncertainty, poor parameter estimation, or other external disturbances. However, in order to control the whole-body behavior of robots with high DOFs in an intuitive way, researchers concentrate on the most important aspects of an element in the robot, and then describe their desired dynamics in the operational space. One of the most widely used approach in operational space is developed by [Khatib 1987], in which the operational space controller is used to compute the necessary generalized forces to achieve desired trajectories generated for task end-effectors. [Khatib 2004b, Sentis 2005] synthesized the robot whole-body behaviors simultaneously by proposing a more specific control frame in the operational space for high-DOFs legged robots with prioritization of tasks using dynamically-consistent null-space projections. They designed a control hierarchy among multiple behavioral primitives with three levels: constraints (such as contacts, joint limits, collisions avoidance, balance control), operational tasks (manipulation and locomotion) and postures (residual motion). These tasks are treated as independent control entities. This approach integrates the contact constraints in the control frame with highest priority and projects the operational tasks into its constraint null-space. Then the posture motion should control the residual movement redundancy without violating the higher tasks [Sentis 2007]. This approach is appealing since it provides a method to monitor multi-behaviors and accomplish them feasibly at runtime. By defining the task priorities, the high-priority tasks are ensured to execute accurately and the less important ones are only excuted as well as possible, without conflicting the high-priority tasks. [Park 2006] further enables the operational space dynamics control by combining contact condition into the formulation, and enables the required torque for all tasks to be consistent with the contact constraints both in the velocity and acceleration levels. [Sentis 2010] later extended this work for control of internal forces, and the unified control of multi-contact constraints, centroidal motion maneuvers, operational tasks, postures. [Hutter 2013b] brought these control methods to simulate a quadruped robot on challenging terrain. These approaches can be extended to operate more compliant tasks by adding error based feedback controllers [Dietrich 2012]. Furthermore, [Mistry 2010] proposed to apply orthogonal decompositions and projections to resolve the inverse dynamics of legged robots and address the estimation of contact forces. This method avoids the

inversion of the mass matrix and gains robustness against model uncertainty. Although the resolution formulations relating to the contact-consistent prioritized control framework are different, it has been demonstrated that they are in fact equivalent, and they produce the same control commands to the same desired accelerations [Righetti 2011b]. Authors in [Righetti 2011a, Righetti 2013] propose to distribute the contact forces in three directions, and the generated torque command allows to minimize tangential contact forces inside the frictional boundaries while guaranteeing the tracking performance of the motion references. However, the analytic solutions are not suitable to handle inequality constraints, such as joint limits, friction cones, obstacle avoidances. The optimization based solutions are detailed in Section 1.2.7.

There are two main formulations to compute the robot dynamic model, including the Lagrange method and the Newton-Euler formulation. The Lagrange approach derives the equations of motion in a systematic way which is independent of the reference coordinate frames without calculating internal forces between elements forming the structure [Sciavicco 2014]. Although Lagrange formulation is conceptually systematic and relatively simple, it is computationally expensive and not suitable for complex system with high DOFs. The Newton-Euler approach provides a more efficient way by exploiting the open structure of one kinematic chain, which directly uses the Newton's and Euler's equations for each body in the chain. This method is especially preferable for the development of efficient recursive algorithms for dynamics-model computations in the joint space, including the recursive Newton-Euler algorithm (RNEA) for the inverse dynamics model [Luh 1980], the articulated-body algorithm (ABA) for the forward dynamics [Featherstone 1983], the composite-rigid-body algorithm (CRBA) for calculating joint-space mass matrix [Walker 1982], the projected Newton-Euler algorithm (proNEu) [Hutter 2013a], and their modified versions for robots with floating bases [Featherstone 2014]. To calculate the dynamics in the operational space, many efforts have been concentrated on the efficient computations of the operational-space inertial matrices, and the dynamically consistent pseudo-inverse of Jacobians and bias forces [Wensing 2012, Wensing 2015].

### 1.2.6 Centroidal Dynamics

As a natural behavior, humans and animals apply whole-body motion to keep balance. [Herr 2008] showed the human neuron-control systems coordinated inter-segmental momentum cancellations, and their centroidal angular momentum are close to zero. This coordination strategy in human subjects has appealed the robotics researchers to pursue its possible applicability to robots, in which the effective total mass of the whole system occupies an important place in the robot dynamics. It is also the uniquely important point in robot's dynamics where the robot's aggregate translational momentum and angular momentum are naturally defined and the resultant gravity forces acts on, so that the control of the angular momentum will lead to natural whole-body behaviors for postural balance [Goswami 2004, Lee 2012]. All external forces on the robot, gravity force and interaction force/moment (such as contact forces of feet or torso with the environment) can be projected on this point, and the rate of change of linear and angular momentums at the CoM is equivalent to resultant effect of all external forces [Featherstone 2014], as shown in Fig. 1.11. The robot centroidal momentum sums each individual link momentum expressed at the robot CoM by using spatial notations [Orin 2008]. The centroidal dynamics is the derivative of the centroidal momentum [Orin 2013], which is further extended by a coordinate transformation that achieves a decoupled equation structure of a robot motion with a floating base [Garofalo 2015].

Control of the robot CoM point is also one example of the operational space control [Khatib 1987], which has been used on floating-base robots to minimize the centroidal angular momentum. Most of the applications using centroidal dynamics are on humanoid robots by combining other approaches. [Ugurlu 2010] implemented the Eulerian ZMP Resolution method to generate the bipedal walking pattern by including an intrinsic angular momentum model. [Ott 2011] regulated the CoM position and the base orientation of a humanoid robot to recover posture by computing a desired wrench from external perturbation, which is based on the contact-force optimization. [Pucci 2016] proposed turn the gain automatically in the momentum based balance controller. [Nava 2016] avoided the zero-dynamics instability by modifying the closed-loop momentum-based controller structure, to allow the asymptotic stability and is shown by Lyapunov analysis on the linearized system's joint space. [Dafarra 2016] implemented a con-

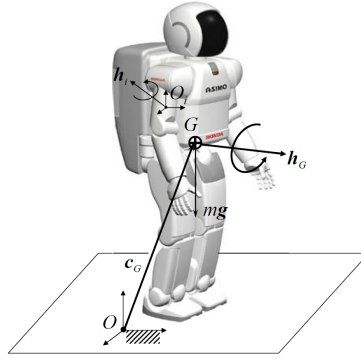


Figure 1.11: Centroidal Dynamics [Orin 2008]

trol strategy based on the Capture Point concept and applied one momentum-based torque controller to generate motion references to provide the iCub robot with push-recovery and reaction abilities to external disturbances. [Traversaro 2017] extended the concept of centroidal dynamics by investigating the role of the base frame in free-floating mechanical systems, and showed how it can be exploited to unify the centroidal dynamics and the equations of motion expressed in the base frame. [Herzog 2014, Herzog 2016] apply one hierarchical torque control framework to solve the multi-task inverse dynamics with inequality constraints, and use a simple linear quadratic regulator (LQR) to compute a linear optimal feedback for momentum control. Moreover, [Wensing 2013b, Wensing 2013a, Wensing 2014b] generate dynamic running, steering, jumping based on one 3D SLIP template through task-space control with conic optimization. [Mason 2014] proposed an experimental evaluation of one LQR approach using a linearization of the full robot dynamics together with the contact constraints. [Koolen 2016] used the momentum-based control on a humanoid robot, Atlas, by applying quadratic programming with constraints from joint accelerations, unilateral ground contacts and grasp-force limits. In addition, this method was extended to quadruped robots. [Mastalli 2016] applied the centroidal dynamics model to compute a robot-feasible CoM trajectory by mapping the contact forces. [Ugurlu 2015] used one virtual admittance control of the rate of change of angular momentum to keep postural balance. However, the applications on legged-on-wheel robots are neglected in the robotics field, and the robot centroidal properties can be added to achieve more autonomous motions. To control the centroidal momentum for whole-body balance in an efficient way, recursive algorithms for centroidal momentum matrix and its derivative are developed in [Orin 2013]. [Wensing 2016] simplified the derivations from the joint-space mass matrix, Coriolis and Centripetal force, which are applied to perform kicking motion by generating the emergent humanoid upper-body motion under the active control of the robot centroidal angular momentum.

### 1.2.7 Optimization based Torque Control

Different with the pseudo-inverse dynamics approach, researchers use optimization techniques to solve the actuated-joint torque which can maneuver more general constraints with inequality and equality forms. This method can be built in the joint-coordinated space or the operational space. The most popular approaches use Quadratic Programming (QP) when there are inequalities such as friction cones, force unidirectionality, joint limits or other constraints, which can be given as follows,

$$\begin{aligned} \max_{\nu} \quad & \|\ddot{e}^* - \ddot{e}\|_N^2 \\ \text{s.t.} \quad & A\nu + \beta = 0 \\ & D\nu + \gamma \geq 0 \end{aligned} \tag{1.2}$$

where  $\nu$  is the control variable,  $A$  and  $\beta$  are matrices which define the suitable equality constraints given by the dynamics model and kinematic non-holonomic contact constraints like  $\dot{x}_c = J_c \ddot{q} + \dot{J}_c \dot{q} = 0$ .  $D$  and  $\gamma$  are matrices that define inequality limits, such as the friction limits. The optimization variable  $\nu$  can contain the generalized torques, the joints' accelerations and the contact forces.

Two QP control schemes can both handle several tasks and constraints simultaneously, including the weight based scheme and the strict hierarchical scheme. The weight-based schemes proposed by [Bouyarmane 2011, Salini 2011] combine the tasks in one QP scheme by distributing weights to each task to achieve whole-body motion control. However, the weighted tasks in this scheme may be in conflict with each other when the robot redundancy is not enough, resulting that no task can be completely fulfilled but achieved with a compromise [Ponce 2014]. Another popularly used solution is based on the hierarchical control schemes which apply several hierarchies to divide the multiple tasks, ensuring that lower-priority hierarchical tasks cannot interfere with high-priority hierarchical ones. In this way, the top-priority hierarchical task is solved firstly and serves as the constraint for the next-following priority task, until the lowest hierarchy task, which means that the multi-hierarchies are treated recursively [Kanoun 2011]. [Saab 2013, Escande 2014] proposed the Hierarchical Quadratic Program (HQP) solver that improves the traditional cascade of QPs to handle the equality and inequality constraints at any hierarchy of the cascade QPs. These control frames with a slack of QPs have been applied on a real torque-actuated humanoid robot (the biped part of the Sarcos Humanoid) [Herzog 2013, Herzog 2014] accounting for multiple constraints and tasks by efficient algorithms that consider constraints recursively. [Righetti 2012, Righetti 2013] minimized any combination of linear and quadratic costs by developing an inverse-dynamics-based torque controller for robots with floating bases, which also respect contact constraints to track each task motion reference. [Ponce 2014] generated sitting, walking, climbing, yoga motions on rough terrain by applying Stack of Tasks (SoT) and HQP that enables to change the task number at any control loop. A prioritization scheme is proposed in [Wensing 2014a, Wensing 2013a] which is based on a conic optimization to control the humanoid centroidal angular momentum to generate movements like kicking and jumping motions. [Hutter 2013a, Hutter 2014] proposed to use the least square optimization to resolve operational-space tasks at multiple operational-space end-effectors by combining the inverse dynamics based prioritized kinematic projections in a hierarchical torque control framework, including the control of the CoG motion, base postures, end-effector positioning, joint-torque and contact-force optimization. [Mastalli 2016, Focchi 2017] applied HQP on the quadruped robot HyQ to achieve the contact-force optimization and the centroidal trajectory optimization with no-slipping constraints. [Hopkins 2015b, Hopkins 2015a] demonstrated push recovery and compliant walking of a biped robot THOR, by using efficient QP to compute optimal joint torques by combining the centroidal dynamics model, frictional contact constraints and joint limits. [Koolen 2016] presented a momentum-based framework for the humanoid robot Atlas, which lies on a quadratic program to reconcile the joint position/velocity/torque limits, unilateral ground contact constraints and grasp-force limits. The optimization based whole-body control has substantially emerged during the DRC competition [Kuindersma 2016] which resulted in a collection of optimization algorithms based on quadratic programming for dynamic planning, control, state estimation for a humanoid robot Atlas to operate reliably from footstep placement to whole-body behaviors, such as the active set based QP [Kuindersma 2014], mixed-integer convex optimization [Deits 2014], two-stage optimization control [Feng 2016], sparse nonlinear optimization [Dai 2014]. Pseudo inverses are integrated in HQP to optimize hybrid tasks relating to motion and force for a quadruped robot [Bellicoso 2016] in which the cascade of QPs embody fewer equality constraints. Moreover, [Herzog 2016] proposed to avoid equality constraints which are generated between hierarchies, by using null-space projectors in the cost function and inequality constraints. Authors in [Dietrich 2012] dealt with safety as the primary priority and enabled the robot to react to various constraints, including contact situation, collision avoidance, self-collision avoidance, feet and torso locomotion tracking, singularity avoidance, torso posture control and mechanical end stops. In addition, the constraint-based Linear Quadratic Regulator (LQR) can also be integrated in quadratic programming to optimize whole-body behaviors around key postures [Yamane 2012, Mason 2014, Mason 2016, Herzog 2016, Kuindersma 2016]. However, the number of the constraints in developed HQPs increases as the hierarchy index increases, due to that the generalized dynamics model itself cannot embody the multi-hierarchy structure and the extra more and more constraints are added to ensure task hierarchy. Moreover, [De Lasa 2010] combined prioritized and weighting-based schemes together to treat hybrid optimization of hierarchical tasks and distributed weights to sub-tasks in the hierarchy using the weighted scheme. The existed hybrid control schemes also present the issues in the pure weighted scheme and the pure hierarchical scheme.

### 1.2.8 Multi-Task Active Compliance

Naturally, animals adapt their leg muscle stiffness for controlling their compliance to adapt their motion to changing terrains [Hoyt 1981]. It will be a big step to close the gap between legged robots and their counterparts by imitating this adaptive capability. When robots operate in unstructured environments and the knowledge of the environment is imprecise, contacts may occur between the robot and the ground, obstacle avoidance is necessary or self-collision possibilities exist. Thus, some kind of interaction control is required essentially to comply with forces [Boaventura 2013a, Focchi 2013]. Compliance in a robotic system can be achieved by passive compliance and active compliance. Passive compliance is achieved using physical springs and dampers between the robot and ground surface to control the interaction forces. However, physical components fix the stiffness and damping, which cannot adapt to the changing environment due to their low frequency of response. Physical components cannot be placed anywhere on the robot, especially for operational-space tasks. In contrast, active compliance uses virtual springs and dampers whose stiffness and damping can be controlled with flexibility to adapt to the geometry with different surface properties. Virtual components can be placed anywhere on the robot, which represents a big improvement of the versatility and safety performance. High-bandwidth impedance control ensures the tracking compliance and outputs sufficiently fast to respond to the impact forces with the environment. Active control method is a good way to emulate virtual compliance both at the joints and operational tasks. With the development of computing capabilities and actuation bandwidth, active compliance can be performed feasibly for highly dynamic applications [Boaventura 2012]. [Boaventura 2013b] performed an experimental study and compare the active and passive compliance, and shows that systems with active impedance can emulate the passive elements in the dynamic range required for the robot locomotion and interaction in the environment, without relevant difference. There are many active compliance-control methods to control the interaction forces, including the impedance control and admittance control [Hogan 1984], hybrid position/force control [Raibert 1981], virtual model control [Pratt 2001], operational space control [Khatib 1987], artificial repulsion potential fields [Khatib 2004a]. An overview of different force control schemes can be found in [Fernández 2010]. The classical force control and hybrid position/force control directly regulate contact forces to desired references [Raibert 1981]. However, desired force references are not easy to be achieved. Virtual model control is an intuitive approach to control robot locomotion behaviors, which regards only the relative elements of the robot to obtain the low-dimension Jacobian, without using inverse dynamics. Virtual components can be placed such as between the hip center and foot tip, and virtual forces can be mapped back to legs' torques by the transpose of the Jacobian. This enables to bring SLIP approach to robot dynamics. It can also be used between the floating base and the environment, and these virtual forces and moments can be mapped back to joints as feed-forward torques of the actuators [Winkler 2014].

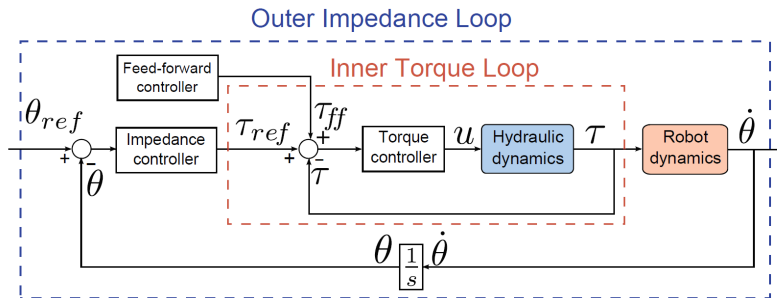


Figure 1.12: Cascade Impedance Controller [Semini 2016]

The impedance control belongs to the family of indirect force control method, and a proper disturbance response can be designed when a deviation motion occurs. It highlights that two physical systems must be physically complementary with each other during dynamic interactions, if one system is an impedance, the other must be an admittance, and vice versa [Part 1985]. [Albu-Schaffer 2002, Ott 2010] give an overview of different impedance control architectures, including impedance controller (stiffness



controller) and admittance controller. They both use a single and common control law to manage simultaneously both force and position by specifying a dynamic relationship (mechanical impedance) between them. Impedance controller measures the deviation of positions and outputs forces, while admittance controller measures the deviation of forces and output positions. Virtual components like stiffness and damping are added between the function points and the environment, which have flexibility to be controlled and adjusted in software depending on environment properties. The dynamic characteristics can be defined at joint space, or any point of the robot in the operational space with the requirement that a kinematic transformation exists between this point and the actuated joint variables, such as the robot end-effectors or the robot center of mass (CoM) [Ott 2008]. Impedance control has been widely used for human-exoskeletons for rehabilitation treatment [Lubecki 2013, Bingquan 2014, Van Dijk 2015, Jinfu 2015, Gong 2016, Caputo 2015]. It provides the assistive torque to human subjects according to the synchronous gait reference. Different level of assistance can be obtained flexibly by providing different virtual impedance. It refers to the human-in-charge mode when the impedance is set to be zero, then the robot will follow the human motion and minimize the interaction force between them. It refers to the robot-in-charge mode when the impedance is configured to be high, the robot moves like in the position control mode and tracks the trajectory reference. The robot behaves in assistive mode when the impedance is set to be low, assistive forces are generated to supply as much interaction force as the human needs to fulfill the training tasks and the maintenance of a normal stepping motion [Riener 2005, Veneman 2007, Banala 2009, Ekkelenkamp 2005, Duschau-Wicke 2010, Cao 2014]. Impedance control is also widely applied in legged robots to interact compliantly with the environment especially for locomotion and grasping functions, including humanoid robots DLR Lightweight Robots [Albu-Schäffer 2007, Ott 2012], THOR [Hopkins 2015a], CoMan [Dallali 2011], WALK-MAN [Tsagarakis 2017], Justin [Dietrich 2016], TORO [Englsberger 2014] and quadruped robots Little dog [Buchli 2009], HyQ [Boaventura 2015, Focchi 2016] with impedance control for hydraulic actuators and electrical motors, HyQ2Max [Semini 2017], Star $LETH$  [Hutter 2012], ANYmal [Hutter 2016], MIT cheetah robot [Seok 2013]. These robots employ the similar cascade impedance control framework, shown in Fig. 1.12. The inner loop is a torque controller like PID and the outer loop is composed by a feedforward controller (such as inverse dynamics control) and a feedback controller (impedance control). Another reactive compliant control strategy is to integrate artificial repulsion potential fields in impedance control [Khatib 1986, Khatib 2004b], which can be applied to perform multiple tasks simultaneously in a pseudo-inverse whole-body control framework [Sentis 2005], or hierarchical QP based torque solvers [Escande 2014, Ponce 2014]. The approach in [Dietrich 2011a] deals with safety as the primary priority and allows the robot to react to constraints in unpredictable, non-modeled dynamics and environments (including collision avoidance, self-collision avoidance, Cartesian impedance at the robot feet and torso, singularity avoidance, torso posture control, mechanical end stops), whose priorities follow intuitive rules from high to low are safety, physical constraints, task execution, and posture adjustment [Dietrich 2012][Dietrich 2011b]. However so far, multi-task active compliance has not been integrated into prioritized operational-space control scheme when various impedance controllers are defined. In the state of the art, for each task reference, the specific feedback impedance controller can be added to enable the compliance on the specific task. However, the general impedance controllers which indirectly influence the whole-body motions are not prioritized and influence all designed tasks together. In this thesis, this problem is tackled by designing a whole-body torque control strategies to manage multi tasks with compliance and priorities.

### 1.3 Approach and Contribution

The long-term target is to develop a quadruped-on-wheel robot that can perform hybrid gaits from rolling-walking, dynamic rolling-trotting, rolling-trotting, etc. In this thesis, we will investigate and research the efficient motion generation approaches for the rolling mode of Towr $ISIR$ , and for the trotting mode of Quad $ISIR$ , in which the robot-self centroidal properties will be researched and integrated. The two locomotion modes serve as the basis for our long-term perspective. It is essential to integrate system design, motion generation, and torque control tightly connected to achieve various highly dynamic

behaviors in unknown environment.

Generally, the approaches used in this thesis relate to spatial vectors, whole-body kinematics, operational space control, null-space projections, hierarchical torque control, generalized and centroidal dynamics, active impedance control, convex optimization. These techniques already exist in the state of the art. Some of them are improved and we propose new concepts to achieve the dynamic rolling locomotion for *TowrISIR* and the legged trotting locomotion for *QuadISIR*. Especially, we take advantages of the centroidal properties of the two robots and propose new motion generators. Then we improve active compliance technique and four kinds of torque controllers which are used to verify the motion generators.

#### **Contributed Torque Control:**

For the torque control, we propose the prioritized impedance controller and integrate this strategy into four improved hierarchical torque control methods. In the pseudo-inverse dynamics approach, the generalized dynamics model is modified to integrate our prioritized impedance controller, which result in a multi-task torque control naturally with the hierarchical compliance. In the null-space weighted convex optimization method, the modified dynamics model is used to achieve the task acceleration extraction. In this way, the prioritized impedance controller is integrated. Compared with conventional weighted QP, the null-space based weighted scheme enables the impedance controllers to be decoupled to the specific designed sub-tasks, especially for impedance controllers which indirectly influence the designed tasks. In addition, for the torque controller based on the hierarchical convex optimization, we propose a new dynamics model to incorporate our prioritized impedance controller. To be emphasized, the new dynamics model embodies the multi-level control hierarchy, the hierarchical task reference is extracted from this model which enables various impedance controllers to be integrated compatibly with priorities. The inequality constraints are modified to be compatible with the optimization solver by extracting the quadratic forms of the contact force and the joint acceleration from our new dynamics model. Then we use our proposed model to integrate the prioritized impedance controller into a more complex scenario which combines the hierarchical and the null-space based weighted schemes together. The modified or new dynamics model enable the achievement of the more general and complete hierarchical optimization solvers, both for the task acceleration extraction and the same number of constraints. In the physical simulations, we test the pseudo-inverse dynamics and the optimization based techniques. The results show that the control schemes with the prioritized impedance controller enable our robot to handle complex missions efficiently and compliantly. These results can also be referred in our published papers [Du 2020a, Du 2020f] and one submitted paper [Du 2020d](Under Review).

#### **Model based Wheeled Locomotion:**

For the rolling-mode locomotion, we propose two motion generators for a quadruped-on-wheel robot, *TowrISIR*, including one wheel-motion generator and one new whole-body motion generator which both track the centroidal motion using the robot rolling-mode locomotion. The whole-body inverse kinematics model is decomposed into components relating to the base motion, the legged motion and the wheel motion. In wheel motion generator, we assume that the desired legged suspension is already achieved, and the wheel motion is derived by combining the centroidal momentum & dynamics model and the whole-body kinematics model. In the second whole-body motion generator, the desired legged suspension can be generated on-line. The centroidal motion relative to the base frame along  $z_b$  direction is controlled using a proposed altitude control model to enable the robot to run on terrains with a much-difference altitude. Then the whole-body motion generator is established by combining the whole-body inverse kinematics model, the wheel-center motion model, the altitude control model and the centroidal momentum/dynamics model. By using our prioritized impedance controller based torque controller in Chapter 3, the contact constraints are ensured and our robot is successfully driven to track the centroidal motion reference and behave compatibly on various simulated rough terrains, which validates the adaptivity and effectiveness of our two motion generators. The wheel motion generator was published in the paper [Du 2020e] and the whole-body motion generator was submitted [Du 2020c](Under Review).

#### **Model based Legged Locomotion:**

For the legged locomotion, we propose a new legged locomotion principle and a legged motion generator for a quadruped robot, *QuadISIR*. The robot leg is newly designed with parallel mechanisms to enable the robot with more symmetric mass distribution and dynamics properties and the generalized dynamics model is built. Then the centroidal momentum and dynamics models are established using each leg centroidal motion instead of each leg-body motion, and each leg centroidal properties are derived. The motion generator is proposed which is much closer to real quadruped locomotion principle. The stance legs follow the base motion using contact-constraint consistent torque controller. The swing legs compensate the influences from the stance legs to the centroidal motion. The desired legged position, velocity and acceleration are achieved using the mass distribution model, the modified centroidal momentum and dynamics models, as well as the kinematics contact constraints. By combining the whole-body kinematics model and the kinematics constraints relating to the contact constraints and the parallel mechanism, our new locomotion principle and generator enable the robot to trot and walk without pre-defined legged-motion trajectories. We also propose a new state machine for switching the locomotion gaits between walking and trotting. This state machine also serves the torque controller which determines the dimension of the legged control states and the contact Jacobian. Therefore, our new locomotion principle integrates the system design, motion generation and torque control together which synthesizes several locomotion gaits using only one control framework. The validation is performed through advanced numerical simulations using our quadruped robot, *QuadISIR*. This work has been submitted on International Journal of Robotics Research [Du 2020b](Under Review).

## 1.4 Thesis Outline

This thesis contains four main chapters (except the presented introduction in this chapter and the conclusion in Chapter 6), namely: Chapter 2: System modeling, Chapter 3: Prioritized impedance force based hierarchical torque control, Chapter 4: Model-based wheeled locomotion, and Chapter 5: Model-based legged locomotion. The quadruped-on-wheel robot *TowrISIR* is employed in Chapter 2, 3, 4. The quadruped robot *QuadISIR* is applied in Chapter 5. The outline of the main contents are showed as follows:

- The introduction is followed by Chapter 2 in which we describe the system-modeling. Initially, the kinematics and dynamics models are established for the quadruped-on-wheel robot, *TowrISIR*. We use this robot to review the modeling approaches in this thesis, including the tools used to build the whole-body kinematics model and the generalized dynamics model, spatial vectors, operational space control, centroidal dynamics, null-space based prioritized control, contact-constraint based hierarchical control. In this chapter, the inverse kinematics model of *TowrISIR* is decomposed into components relating to the base motion, as well as the legged and wheeled motions, which will be applied in the wheel motion generator in Chapter 4. In addition, we derive the centroidal dynamics model in a more apparent way, and propose a set of more compact formulas for multi-task operational-space torque control that can extract one prioritized task acceleration in a recursive way. This chapter serves as the basis for the following three chapters which constitute our main contributions.
- In Chapter 3, we improve four kinds of torque controllers, including a pseudo-inverse based dynamics method and three improved optimization-based techniques: a null-space based weighted optimization solver, a hierarchical optimization solver, and a hybrid hierarchical-weighted solver. We also propose one concept called prioritized impedance controller, which enables various impedance control forces to be consistent and compatible with multi-task hierarchies. We propose a modified model for the first two torque controllers, and a new dynamics model which embodies the multi-level control hierarchy to improve the latter two torque controllers to be more general and complete hierarchical optimization solvers, both for the task acceleration extraction and the same number of constraints. We implement these control schemes on our virtual quadruped-on-wheel robot to test the usefulness and validity of our approaches.



- In Chapter 4, we focus on the rolling mode and two model based motion generators are proposed which take advantages of the centroidal properties. In the first wheel motion generator, the desired legged motion is assumed to be known, and the desired wheel motion is achieved by combining the whole-body kinematics model at the velocity and acceleration levels, with the centroidal momentum & dynamics models. In the second whole-body motion generator, the desired legged and wheel motions are obtained by combining both the kinematics model, the wheel-center motion model, the robot centroidal momentum and dynamics models, and a proposed altitude control model. The two motion generators enable the robot to run on various terrains, especially on terrains with much altitude difference by the whole-body motion generator. They are verified by the proposed torque controller in Chapter 3 using several numerical simulations.
- In Chapter 5, the quadruped robot, *QuadISIR* is used and the main contribution concentrates on a novel locomotion principle and a new legged motion generator, without leg-massless assumption or pre-defined leg trajectory. Our quadruped robot is newly designed with symmetric dynamics properties using parallelogram mechanisms, and the corresponding generalized dynamics model is built. Then the centroidal properties of the whole robot and each leg are further extended and analyzed, in which the virtual joint between the base and each leg CoM is used. By combining the whole-body kinematics model and the kinematics constraints relating to the contacts and the parallelogram mechanisms, the adaptive swing-leg motion is achieved online during the running process. We also propose a new state machine for switching the locomotion gaits, which also serves the torque controller which determines the dimension of the legged control states and the contact Jacobian. We apply our hierarchical torque controller to enable the robot with compliant dynamics properties by respecting the kinematics constraints of the non-slip contacts and the parallelogram mechanisms. The robot is simulated to trot in forward and lateral directions, turn, handle push-recovery motion, as well as trot on relatively rough terrains.
- The Chapter 6 closes the thesis with conclusions and future perspectives.
- Two appendixes are shown by following the conclusion chapter, including the Appendix A: Robot Description, Appendix B: Simulation Environment.

# System Modeling

---

## Contents

<b>2.1 Floating-base Multi-Body Dynamics of Towr<i>ISIR</i></b> . . . . .	<b>21</b>
2.1.1 Mechanism Structure & Spatial Notations . . . . .	22
2.1.2 Wheel Contact Point Estimation . . . . .	26
2.1.3 Whole-body Inverse kinematics of Towr <i>ISIR</i> . . . . .	26
2.1.4 Generalized Dynamics Model . . . . .	28
2.1.5 Floating-base Centroidal Dynamics . . . . .	29
<b>2.2 Operational Space Torque Control</b> . . . . .	<b>31</b>
2.2.1 General Control Framework . . . . .	31
2.2.2 Operational Space Control (OSC) . . . . .	31
2.2.3 Contact-constraint Consistent Projections . . . . .	33
2.2.4 Contact Constraint consistent Prioritized OSC . . . . .	34
<b>2.3 Summary</b> . . . . .	<b>35</b>

---

In this chapter, we will build the dynamics model for the quadruped-on-wheel robot, Towr*ISIR*. It belongs to the class of floating-base systems, its torso is not rigidly connected with the environment and the whole system can move freely in space using the ground as support. This kind of robot unifies strengths of legged robots and wheeled rovers and can possess more locomotion gaits, but meanwhile proposing higher DOFs and control difficulties. This system has an underactuated floating base and can be propelled forward with wheel rolling, dynamic legged locomotion by changing contact points, or combination of hybrid locomotion modes. The system model of Towr*ISIR* in this chapter is also used in the following two chapters using the rolling mode, and the approaches will also be applied in Chapter 5 for another quadruped robot with parallelogram mechanisms to verify the dynamic legged locomotion. The hybrid rolling-walking, rolling-trotting modes will be researched in the future work.

This chapter provides the theoretical background to model the kinematics and dynamics of the multi-body system. First, we review the spatial vectors [Featherstone 2014] which are used to build the inverse kinematics and dynamics model of Towr*ISIR*. The inverse kinematics model is decomposed into components relating to the base motion, the legged and wheeled motions, which serve as basis in the wheel motion generation in following chapters. Then we derive the centroidal dynamics model in another way which is different with that in the state of the art. Secondly, we review the operational space control (OSC) and introduce three kinds of projections relating to the contact constraints [Khatib 1987, Sentis 2007]. Finally, the formulations of the contact-consistent prioritized OSC are introduced which are further rewritten in a more compact recursive way.

## 2.1 Floating-base Multi-Body Dynamics of Towr*ISIR*

“Towr*ISIR*” is a new quadruped-on-wheel robot that is designed in ISIR laboratory (see Fig. 2.1), equipped with 12 leg-revolute joints and 4 wheels. The robot mass is about 100kg 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. 2.2. We select the same hydraulic actuators with HyQ2Max [Semini 2017] and the power is enough for our robot to handle our designed missions in next two chapters. Compared

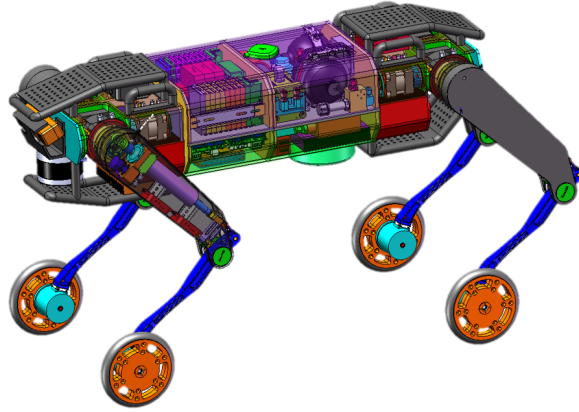


Figure 2.1: “TowerISIR”—A quadruped-on-wheel robot developed in ISIR lab is equipped with 12 hydraulic actuators and 4 wheel motors. Each leg has 4 joints, including the hip abduction/adduction joint (HAA), the hip flexion/extension joint (HFE), the knee flexion/extension joint (KFE) and the wheel forward/backward joint (WFB). The joints with hydraulic actuators apply the similar structure with the robot HyQ2Max [Semini 2017] which has different leg layout and no wheels. Each joint is integrated with the force/torque sensor and the absolute/incremental encoder.

with conventional quadruped robots, the four wheel joints induce more complexity into the kinematics and dynamics models, which will be detailed in this chapter. In addition, we derive the whole-body dynamics model without any leg massless assumptions. The whole-body inverse kinematics model is derived as the basis for the whole-body motion generator in other chapters. We concentrate on the rolling mode and all wheels are always maintained in contact with the ground. In this chapter, spatial vectors [Featherstone 2014] 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 chapter, the subscript  $i \in [1, \dots, 4]$  always denotes the body/joint index, and the  $i$  index in the subscript  $r_i$ ,  $k_i$ ,  $c_i$  and  $w_i$  denotes the leg index.

In this section, we derive the inverse kinematics, contact-point estimation and generalized dynamics model after the review of the spatial vectors. Then the centroidal momentum/dynamics model is derived in a more apparent way, different with the derivation process in [Orin 2013, Wensing 2016].

### 2.1.1 Mechanism Structure & Spatial Notations

The robot and mechanism design are based on requirements that multiple kinds of locomotion gaits can be achieved. For each leg of quadrupedal robot, three degrees of freedom (DOF) are necessary at least that allow the forward, lateral and vertical movements which are essential for versatility. Under each foot, one wheel is connected to the lower leg which adds the rolling mode to the whole system. Due to the outstanding steering ability of the quadrupedal robot, each wheel module has only one DOF which eliminates the steering revolute joint at each wheel. The whole kinematic structure of this robot is shown in Fig. 2.2 with 16 DOFs. Namely, each leg contains three revolute joints and one wheel joint, namely hip abduction/adduction joint (HAA), hip flexion/extension joint (HEF), knee flexion/extension joint (KFE) and wheel forward/backward joint (WFB). The torso is treated as the floating base which has one 6-DOF unactuated joint respecting to the inertial frame. Compared with conventional quadruped-on-wheel robots such as Hylos and Hylos II, legs’ configuration in TowerISIR is changed from “O”-configuration to “M”-configuration, one hip abduction/adduction joint replaces the wheel steering revolute joint, which decreases the mass of each end-effector and reduces the torque burden at hip and knee joints. From the robot structure in Fig. 2.1 and its mechanism in Fig. 2.2, the joint configuration is defined as follows,

$$\mathbf{q} = [\mathbf{x}_b^T \quad \mathbf{q}_g^T]^T = [\mathbf{x}_b^T \quad \mathbf{q}_{g1}^T \quad \mathbf{q}_{g2}^T \quad \mathbf{q}_{g3}^T \quad \mathbf{q}_{g4}^T]^T \in \mathbb{R}^{n_b+4n_l=n}, \quad (2.1)$$

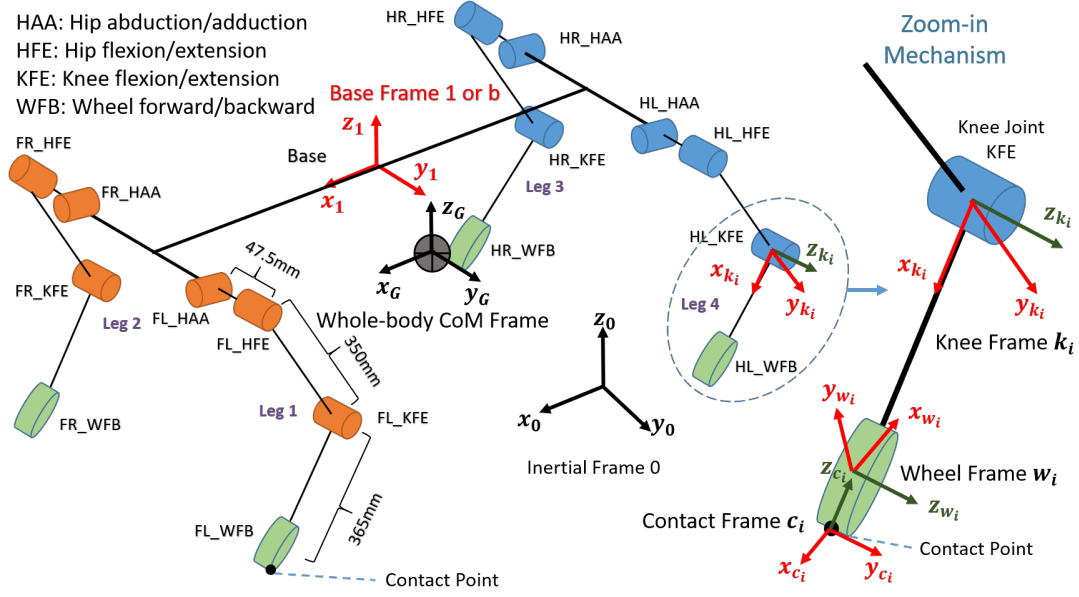


Figure 2.2: Structure of TowerISIR with leg indexes: The whole-body CoM frame  $G$  is set parallel with the base frame 1 or  $b$ . The frames relating to the inverse kinematics are zoomed in, including the contact frame  $c_i$ , the wheel frame  $w_i$  and the knee frame  $k_i$ . The letters F, L, R and H represent front, left, right and hind. On the front-left (FL) leg, we give the leg lengths. The wheel diameter is  $150\text{mm}$ . The contact frame is always estimated at the lowest point in the sagittal plane of the wheel in the gravity direction.

where  $\mathbf{x}_b = [\boldsymbol{\theta}_b^T \ \mathbf{p}_b^T]^T$  is the orientation and position of the floating base in the inertial global coordinate, accounting  $n_b = 6$  degrees of freedom (DOF).  $\boldsymbol{\theta}_b = [\alpha \ \beta \ \gamma]^T$  and  $\mathbf{p}_b = [p_{b_x} \ p_{b_y} \ p_{b_z}]$ . The robot has  $n_l = 4$  legs and each leg has 4 joints.  $n = 4n_l + n_b = 22$  denotes the whole-body DOF.  $\mathbf{q}_{g_i} = [\mathbf{q}_{r_i}^T \ q_{w_i}]^T \in \mathbb{R}^4$  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  $q_{w_i}$ . We define another two vectors to include each  $\mathbf{q}_{r_i}$  and  $q_{w_i}$  separately as follows,

$$\mathbf{q}_r = [\mathbf{q}_{r_1}^T \ \mathbf{q}_{r_2}^T \ \mathbf{q}_{r_3}^T \ \mathbf{q}_{r_4}^T]^T \in \mathbb{R}^{n_r=12}, \quad (2.2a)$$

$$\mathbf{q}_w = [q_{w_1} \ q_{w_2} \ q_{w_3} \ q_{w_4}]^T \in \mathbb{R}^{n_w=4}, \quad (2.2b)$$

where  $\mathbf{q}_r$  include all the leg-revolute joints except for the wheels, and  $\mathbf{q}_w$  denotes all wheel-joint configurations.

TowerISIR consists of  $N_B = 17$  bodies. The ground with the inertial frame is fixed as body 0 and the floating base is indexed as body 1. The other bodies are indexed from 2 to  $N_B$ . Body  $i$ 's parent body is labeled as  $p(i)$ . A coordinate frame is attached to each body to describe the body's motion as a local frame, and the kinematics between neighboring bodies is represented by the connecting joint. The numbering manner is shown in Fig. 2.3: 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, and each one of other joints has only one DOF. We set one frame on each joint, and the frame index has the same index with the joint. 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  $j + 1$ . The set  $c(i)$  is configured as the children set of body  $i$  and  $c^*(i)$  is defined as the body set in the sub-tree which is rooted at the body  $i$ , including the body  $i$  itself. The relationships in Figure 2.3 are shown in Tab. 2.1.

The rotational forward kinematics between adjacent frames throughout this thesis is indicated by the rotation sequences ( $X - Y - Z$ ), each rotation is around the parent frame which is coincident with the configuration of ROS-Gazebo simulation configuration in URDF files. One point position in different

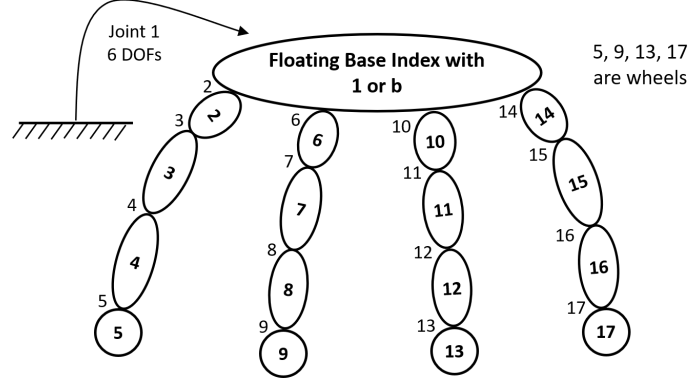


Figure 2.3: Connectivity of TowerISIR with body and joint indexes.

Table 2.1: Parent and children relationships of one body in TowerISIR

Body $i$	$p(i)$	$c(i)$	$c^*(i)$	Body $i$	$p(i)$	$c(i)$	$c^*(i)$
0	N/A	1	1,2,..., 16,17	9	8	N/A	N/A
1	0	2,6,10,14	2,3, ...,16,17	10	1	11	11,12,13
2	1	3	3,4,5	11	10	12	12,13
3	2	4	4,5	12	11	13	13
4	3	5	5	13	12	N/A	N/A
5	4	N/A	N/A	14	1	15	15,16,17
6	1	7	7,8,9	15	14	16	16,17
7	6	8	8,9	16	15	17	17
8	7	9	9	17	16	N/A	N/A

coordinates can be achieved by homogeneous transformation matrix  ${}^j\mathbf{T}_i$  as follows,

$${}^j\mathbf{T}_i = \begin{bmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{p}_i \\ \mathbf{0} & 1 \end{bmatrix}, \quad {}^i\mathbf{T}_j = {}^j\mathbf{T}_i^{-1} = \begin{bmatrix} {}^j\mathbf{R}_i^T & -{}^j\mathbf{R}_i^T {}^j\mathbf{p}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (2.3)$$

where  ${}^j\mathbf{R}_i$  transforms a vector in coordinate  $i$  to a vector expressed in coordinate  $j$  which can be treated as a fixed frame.  ${}^j\mathbf{p}_i$  denotes the body/frame  $i$  position expressed in frame  $j$ .

6-dimensional spatial vectors [Featherstone 2014] give compact notations to describe the robot kinematics and dynamics. In this thesis, all spatial vectors will be written in upright bold letters (e.g.,  $\mathbf{f}$ ,  $\mathbf{v}$ ). By using the spatial notations, the spatial velocity of body  $i$   $\mathbf{v}_i$  with respect to its predecessor body  $p(i)$  is as follows,

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{w}_i \\ \mathbf{v}_i \end{bmatrix} = {}^i\mathbf{X}_{p(i)} \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i, \quad (2.4)$$

where  $\mathbf{w}_i$  and  $\mathbf{v}_i$  are the angular and linear velocities of body  $i$ .  $\mathbf{v}_{p(i)}$  is the spatial velocity of body  $p(i)$ . It is noticed that  $\mathbf{v}_i$  and  $\mathbf{v}_{p(i)}$  are the velocities with respect to their local frames  $i$  and  $p(i)$ .  ${}^i\mathbf{X}_{p(i)} \in \mathbb{R}^{6 \times 6}$  is the transformation matrix of spatial motion vectors from Body  $p(i)$  frame to Body  $i$  frame.  $\Phi_i \in \mathbb{R}^{6 \times n_i}$  represents the free modes of joint  $i$  which is a full-column-rank matrix,  $n_i$  is the number of DOFs relating to the free modes, and  $\Phi_i$ 's orthogonal matrix  $\Phi_i^c$  represents the constrained modes of the joint, so  $[\Phi_i, \Phi_i^c]$  is a basis of  $\mathbb{R}^6$  and is invertible. For revolute joints with only one DOF,  $\Phi_i = [0, 0, 1, 0, 0, 0]$ , and for the floating-base joint with six DOFs,  $\Phi_1 = \Phi_b = \mathbf{1}_{6 \times 6}$  where  $\mathbf{1}_{6 \times 6}$  is the identity matrix. The Jacobian of the floating-base motion is  $\mathbf{J}_b = \mathbf{J}_1 = [\Phi_1 \ \mathbf{0}_{6 \times n}]$ .  $\dot{\mathbf{q}}_i$  represents the joint  $i$ 's joint change rate, for joint 1,  $\dot{\mathbf{q}}_1 = [\mathbf{w}_1^T \ \mathbf{v}_1^T]^T = [\mathbf{w}_b^T \ \mathbf{v}_b^T]^T$ . By using (2.4), the Jacobians for body  $i$ ,  $i \in [1, 17]$  can be derived recursively. The spatial transform matrix  ${}^i\mathbf{X}_{p(i)}$  of spatial vectors

contains the position vector  ${}^i\mathbf{p}_{p(i)} \in \mathbb{R}^3$  and the rotation matrix  ${}^i\mathbf{R}_{p(i)} \in \mathbb{R}^{3 \times 3}$ , as below,

$${}^i\mathbf{X}_{p(i)} = \begin{bmatrix} {}^i\mathbf{R}_{p(i)} & \mathbf{0} \\ \mathbf{S}({}^i\mathbf{p}_{p(i)}) {}^i\mathbf{R}_{p(i)} & {}^i\mathbf{R}_{p(i)} \end{bmatrix}, \quad (2.5)$$

where  $\mathbf{S}({}^i\mathbf{p}_{p(i)})$  denotes a skew-symmetric matrix of the vector  ${}^i\mathbf{p}_{p(i)}$  by satisfying  $\mathbf{S}({}^i\mathbf{p}_{p(i)})\boldsymbol{\omega} = {}^i\mathbf{p}_{p(i)} \times \boldsymbol{\omega}$  for any 3D vector  $\boldsymbol{\omega}$ .  ${}^i\mathbf{p}_{p(i)}$  is the position vector from the frame  $i$  origin to the frame  $p(i)$  origin, which also represents the position of frame  $p(i)$  in the coordinate of frame  $i$ . The relationship of the inverse transformation is expressed as  ${}^i\mathbf{X}_{p(i)} = {}^{p(i)}\mathbf{X}_i^{-1}$ . The spatial inertia of body  $i$  is denoted as  $\mathbf{I}_i$  which is expressed in body  $i$ 's local frame and maps the spatial-motion vectors to the spatial-force vectors, as follows,

$$\mathbf{I}_i = \begin{bmatrix} \bar{\mathbf{I}}_i + m_i \mathbf{S}(\mathbf{c}_i) \mathbf{S}(\mathbf{c}_i)^T & m_i \mathbf{S}(\mathbf{c}_i) \\ m_i \mathbf{S}(\mathbf{c}_i)^T & m_i \mathbf{1}_{3 \times 3} \end{bmatrix}, \quad (2.6)$$

where  $\bar{\mathbf{I}}_i \in \mathbb{R}^{3 \times 3}$  denotes the standard angular inertia at the body  $i$  CoM.  $m_i$  represents the body  $i$  mass and  $\mathbf{c}_i$  represents the body  $i$ 's CoM vector in body  $i$ 's coordinate. If the  $i$ 's coordinate origin is set at the body  $i$ 's CoM,  $\mathbf{S}(\mathbf{c}_i) = \mathbf{c}_i \times = \mathbf{0}_{3 \times 3}$  and the spatial inertia becomes a more compact notation as below,

$$\mathbf{I}_i = \begin{bmatrix} \bar{\mathbf{I}}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & m_i \mathbf{1}_{3 \times 3} \end{bmatrix}. \quad (2.7)$$

In addition, the body  $i$  spatial inertia  $\mathbf{I}_i$  can be expressed in body  $j$  frame as follows,

$${}^j\mathbf{I}_i = {}^j\mathbf{X}_i^{-T} \mathbf{I}_i {}^i\mathbf{X}_j. \quad (2.8)$$

The spatial momentum of body  $i$  can be obtained as below,

$$\mathbf{h}_i = \begin{bmatrix} \mathbf{l}_i \\ \mathbf{k}_i \end{bmatrix} = \mathbf{I}_i \mathbf{v}_i, \quad (2.9)$$

where  $\mathbf{h}_i \in \mathbb{R}^6$  consists of its translational momentum  $\mathbf{l}_i \in \mathbb{R}^3$  and its angular momentum  $\mathbf{k}_i \in \mathbb{R}^3$ . Another important feature of the spatial inertia is to map the change rate of the spatial momentum to the spatial force acting on the rigid body as follows,

$$\begin{aligned} \mathbf{f}_i &= \frac{d\mathbf{h}_i}{dt} = \frac{d}{dt}(\mathbf{I}_i \mathbf{v}_i) = \mathbf{I}_i \mathbf{a}_i + \dot{\mathbf{I}}_i \mathbf{v}_i, \\ &= \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i, \end{aligned} \quad (2.10)$$

The above single formulation incorporates both Newton and Euler equations of a rigid-body motion in which the net force  $i$  is represented in the  $i$ 's local frame. This spatial net force can also be expressed in body  $j$ 's frame using  ${}^j\mathbf{f}_i = {}^j\mathbf{X}_i^{-T} \mathbf{f}_i$ . From (2.4), the spatial acceleration can be represented as follows,

$$\begin{aligned} \mathbf{a}_i &= {}^i\mathbf{X}_{p(i)} \mathbf{a}_{p(i)} + \dot{\Phi}_i \ddot{\mathbf{q}}_i + \ddot{\Phi}_i \dot{\mathbf{q}}_i \\ &= {}^i\mathbf{X}_{p(i)} \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + (\ddot{\Phi}_i \dot{\mathbf{q}}_i + \mathbf{v}_i \times \Phi_i \dot{\mathbf{q}}_i), \end{aligned} \quad (2.11)$$

where  $\ddot{\Phi}_i$  always equals to zero except the universal joint. The spatial cross product operation  $\times$  between motion-motion vectors and between motion-force vectors are given as below,

$$\begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} \times motion = \begin{bmatrix} \boldsymbol{\omega} \times & \mathbf{0}_{3 \times 3} \\ \mathbf{v} \times & \boldsymbol{\omega} \times \end{bmatrix} \times motion, \quad (2.12a)$$

$$\begin{bmatrix} \mathbf{w} \\ \mathbf{v} \end{bmatrix} \times force = \begin{bmatrix} \boldsymbol{\omega} \times & \mathbf{v} \times \\ \mathbf{0}_{3 \times 3} & \boldsymbol{\omega} \times \end{bmatrix} \times force. \quad (2.12b)$$

For example,  $\mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i$  use motion-force cross product operation.

### 2.1.2 Wheel Contact Point Estimation

The kinematics and dynamics models of *TowrISIR* are derived by given the prerequisite that the wheel-contact point and the coordinate set at the point are known. However, compared with normal quadrupedal robots, when interacting with the rough terrain, the wheel-contact point between legged-on-wheel robot and the ground is not easy to be obtained. Since the robot knows nothing about the environment, the wheel contact points can be estimated at the lowest points of the wheels, which are located in the wheel sagittal plane along the gravity direction, as shown in Fig. 2.2. For motion generation, the wheel contact points can be estimated in this subsection by given the floating base and the legged motion, and the wheel-rotation angles detected by sensors.

The wheel-center position can be achieved by given the base position and the legged configuration, we can get the contact position using the normal vector of each wheel median plane and the wheel radius. The normal vector  $\mathbf{z}_{w_i} = [a_i, b_i, c_i]^T$  of wheel median plane in the inertial frame in Fig. 2.2 can be achieved easily which is omitted here. The  $\mathbf{y}_{c_i}$  axis in Fig. 2.2 is parallel with the  $\mathbf{z}_{w_i}$  axis, therefore,  $\mathbf{y}_{c_i} = \mathbf{z}_{w_i}$ . Due to the wheel circle in the 3D space with angles to axes of the inertial frame, another vector  $\mathbf{x}_{c_i}$  parallel with  $\mathbf{x}_0\mathbf{y}_0$  surface should be achieved which is orthogonal to  $\mathbf{z}_{w_i}$ , therefore,  $\mathbf{z}_{w_i} \cdot \mathbf{x}_{c_i} = 0$ .  $\mathbf{x}_{c_i}$  can be expressed as follows,

$$\mathbf{x}_{c_i} = \frac{[b_i \quad -a_i \quad 0]^T}{\|[b_i \quad -a_i \quad 0]^T\|}. \quad (2.13)$$

The axis  $\mathbf{z}_{c_i}$  connecting the wheel center and the contact point is orthogonal with  $\mathbf{z}_{w_i}$  and  $\mathbf{x}_{c_i}$ , therefore,  $\mathbf{z}_{c_i}^T \mathbf{x}_{c_i} = 0$  and  $\mathbf{z}_{w_i}^T \mathbf{z}_{c_i} = 0$ . Then  $\mathbf{z}_{c_i}$  can be achieved as below,

$$\mathbf{z}_{c_i} = -\frac{\begin{bmatrix} \frac{a_i c_i}{a_i^2 + b_i^2} & \frac{b_i c_i}{a_i^2 + b_i^2} & -1 \end{bmatrix}^T}{\left\| \begin{bmatrix} \frac{a_i c_i}{a_i^2 + b_i^2} & \frac{b_i c_i}{a_i^2 + b_i^2} & -1 \end{bmatrix}^T \right\|}. \quad (2.14)$$

The contact position in the inertial frame  ${}^0\mathbf{p}_{c_i}$  can be achieved as follows,

$${}^0\mathbf{p}_{c_i} = {}^0\mathbf{p}_{w_i} - r\mathbf{z}_{c_i}. \quad (2.15)$$

By using the rotation matrix between the inertial frame and the contact frame, the spatial transformation matrix  ${}^0\mathbf{X}_{c_i}$  can be achieved as below,

$${}^0\mathbf{R}_{c_i} = [\mathbf{x}_{c_i} \quad \mathbf{y}_{c_i} \quad \mathbf{z}_{c_i}], \quad (2.16a)$$

$${}^0\mathbf{X}_{c_i} = \begin{bmatrix} {}^0\mathbf{R}_{c_i} & \mathbf{0} \\ \mathbf{S}({}^0\mathbf{p}_{c_i}) & {}^0\mathbf{R}_{c_i} \end{bmatrix}, \quad (2.16b)$$

where  ${}^0\mathbf{X}_{c_i}$  and the adjacent spatial transformation matrices  ${}^i\mathbf{X}_j$  between any two bodies can be used to achieve the Jacobian at the contact points.

### 2.1.3 Whole-body Inverse kinematics of *TowrISIR*

In this thesis, we assume that static friction between all contacts between the wheels and the ground are big enough and all wheels are in pure rolling condition and belong to hard contact constraints whose velocities and accelerations are zero as below,

$$\mathbf{v}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (2.17a)$$

$$\dot{\mathbf{v}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0}. \quad (2.17b)$$

where  $\mathbf{v}_c = [\mathbf{v}_{c_1}^T \quad \mathbf{v}_{c_2}^T \quad \mathbf{v}_{c_3}^T \quad \mathbf{v}_{c_4}^T]^T$  is the translational velocities of all wheel contact points. Each contact frame  $c_i$  is shown in Fig. 2.2 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 [Pucci 2016]. In this subsection, we will decompose the wheel velocities and accelerations out from the contact constraints in (2.17). Firstly, we re-derive the spatial velocity and acceleration of the leg  $i$  wheel contact point respectively using spatial vectors as follows,

$$\mathbf{v}_{c_i} = \begin{bmatrix} \boldsymbol{\omega}_{c_i} \\ \mathbf{v}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i} - \Phi_{c_i} \mathbf{v}_{w_i[z]}, \quad (2.18a)$$

$$\dot{\mathbf{v}}_{c_i} = \begin{bmatrix} \dot{\boldsymbol{\omega}}_{c_i} \\ \dot{\mathbf{v}}_{c_i} \end{bmatrix} = {}^{c_i}\mathbf{X}_{w_i} \dot{\mathbf{v}}_{w_i} - \Phi_{c_i} \dot{\mathbf{v}}_{w_i[z]} - ({}^{c_i}\mathbf{X}_{w_i} \mathbf{v}_{w_i}) \times \Phi_{c_i} \mathbf{v}_{w_i[z]}, \quad (2.18b)$$

where the symbol  $\times$  denotes the cross product for spatial vectors, detailed in [Featherstone 2014].  $\mathbf{v}_{c_i} \in \mathbb{R}^6$  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.  $\Phi_{c_i} = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$  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 [Siciliano 2016]. The subscript  $w_i$  denotes the leg  $i$  wheel frame, shown in the zoom-in mechanism of Fig. 2.2.  ${}^{c_i}\mathbf{X}_{w_i}$  is the transformation matrix for spatial vectors from the wheel frame to the contact point frame.  $\mathbf{v}_{w_i}$  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. 2.2).  $\mathbf{v}_{w_i}$  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  $\Phi_{c_i} \mathbf{v}_{w_i[z]}$  and  $\Phi_{c_i} \dot{\mathbf{v}}_{w_i[z]}$  are zero and have no influence on the translational motion of the wheel contact points. From the translational components in (2.18a) and (2.18b), we can derive the detailed expression of each contact  $\mathbf{J}_{c_i}$  and  $\dot{\mathbf{J}}_{c_i} \dot{\mathbf{q}}$ . In addition,  $\mathbf{J}_c$  and the bias item  $\dot{\mathbf{J}}_c \dot{\mathbf{q}} = [(\dot{\mathbf{J}}_{c_1} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_2} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_3} \dot{\mathbf{q}})^T \ (\dot{\mathbf{J}}_{c_4} \dot{\mathbf{q}})^T]^T$  are used in the contact-consistent prioritized operational space control [Park 2006]. The wheel spatial velocity can be derived as below,

$$\mathbf{v}_{w_i} = {}^{w_i}\mathbf{X}_{k_i} \mathbf{v}_{k_i} + \Phi_{w_i} \dot{\mathbf{q}}_{w_i}, \quad (2.19a)$$

$$\dot{\mathbf{v}}_{w_i} = {}^{w_i}\mathbf{X}_{k_i} \dot{\mathbf{v}}_{k_i} + \Phi_{w_i} \dot{\mathbf{q}}_{w_i} + \mathbf{v}_{w_i} \times \Phi_{w_i} \dot{\mathbf{q}}_{w_i}, \quad (2.19b)$$

where the subscript  $k_i$  represents the leg  $i$  knee joint frame, shown in Fig. 2.2.  ${}^{w_i}\mathbf{X}_{k_i}$  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.  $\Phi_{w_i} = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$  represents the free modes of the leg  $i$  wheel joint. We substitute (2.19a) into (2.18a), then the wheel joint velocity  $\dot{\mathbf{q}}_{w_i}$  can be extracted out by selecting the translational part of (2.18a) as follows,

$$\mathbf{v}_{c_i} = \mathbf{S}_c \mathbf{v}_{c_i} = \mathbf{S}_c {}^{c_i}\mathbf{X}_{k_i} \mathbf{v}_{k_i} + \mathbf{S}_c {}^{c_i}\mathbf{X}_{w_i} \Phi_{w_i} \dot{\mathbf{q}}_{w_i} = \mathbf{0}, \quad (2.20)$$

where  $\mathbf{S}_c = [\mathbf{0}_{3 \times 3} \ \mathbf{1}_{3 \times 3}]$  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 (2.20) and the contact velocity constraint is rewritten as follows,

$$\mathbf{v}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{L} \begin{bmatrix} \mathbf{v}_b \\ \dot{\mathbf{q}}_r \end{bmatrix} + \mathbf{L}_w \dot{\mathbf{q}}_w = \mathbf{0}, \quad (2.21)$$

where  $\mathbf{v}_b = [\boldsymbol{\omega}_b^T \ \mathbf{v}_b^T]^T$  is the base spatial velocity and it 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 below,

$$\mathbf{L} = \begin{bmatrix} \mathbf{S}_c {}^{c_1}\mathbf{X}_{k_1} \mathbf{J}_{k_1} \\ \mathbf{S}_c {}^{c_2}\mathbf{X}_{k_2} \mathbf{J}_{k_2} \\ \mathbf{S}_c {}^{c_3}\mathbf{X}_{k_3} \mathbf{J}_{k_3} \\ \mathbf{S}_c {}^{c_4}\mathbf{X}_{k_4} \mathbf{J}_{k_4} \end{bmatrix}, \quad \mathbf{L}_w = \begin{bmatrix} \mathbf{L}_{w_1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{L}_{w_2} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{L}_{w_3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{L}_{w_4} \end{bmatrix}, \quad (2.22)$$

where  $\mathbf{L}_{w_i} = \mathbf{S}_c {}^{c_i}\mathbf{X}_{w_i} \Phi_{w_i}$  and  $\mathbf{0}_{3 \times 1}$  denotes the zero vector.  $\mathbf{J}_{k_i}$  is the Jacobian relating to the knee joint velocity. In (2.21),  $\mathbf{L}_b$  denotes the left 6 columns of  $\mathbf{L}$ , and  $\mathbf{L}_r$  is the right  $n_r$  columns of  $\mathbf{L}$ . (2.21)



can be decomposed as below,

$$\mathbf{v}_c = \mathbf{L}_b \mathbf{v}_b + \mathbf{L}_r \dot{\mathbf{q}}_r + \mathbf{L}_w \dot{\mathbf{q}}_w, \quad (2.23a)$$

$$= \mathbf{L}_{b_\omega} \boldsymbol{\omega}_b + \mathbf{L}_{b_v} \mathbf{v}_b + \mathbf{L}_r \dot{\mathbf{q}}_r + \mathbf{L}_w \dot{\mathbf{q}}_w = \mathbf{0}, \quad (2.23b)$$

where  $\mathbf{L}_{b_\omega}$  and  $\mathbf{L}_{b_v}$  denote the left and right 3 columns of  $\mathbf{L}_b$  respectively. In this way, (2.23b) serves as the whole-body differential kinematics model for the proposed whole-body velocity generator in Section 4.2.1. Similarly, the contact constraint in the kinematics second order is rewritten as below,

$$\dot{\mathbf{v}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{L} \begin{bmatrix} \dot{\mathbf{v}}_b \\ \ddot{\mathbf{q}}_r \end{bmatrix} + \mathbf{L}_w \ddot{\mathbf{q}}_w + \dot{\mathbf{J}}_c \dot{\mathbf{q}}, \quad (2.24a)$$

$$= \mathbf{L}_b \dot{\mathbf{v}}_b + \mathbf{L}_r \ddot{\mathbf{q}}_r + \mathbf{L}_w \ddot{\mathbf{q}}_w + \dot{\mathbf{J}}_c \dot{\mathbf{q}}, \quad (2.24b)$$

$$= \mathbf{L}_{b_\omega} \dot{\boldsymbol{\omega}}_b + \mathbf{L}_{b_v} \dot{\mathbf{v}}_b + \mathbf{L}_r \ddot{\mathbf{q}}_r + \mathbf{L}_w \ddot{\mathbf{q}}_w + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (2.24c)$$

In this way, (2.24c) serves as the whole-body double-differential kinematics model for the proposed whole-body acceleration generator in Section 4.2.2.

### 2.1.4 Generalized Dynamics Model

When all kinematics, including position vectors and velocity vectors and accelerations are expressed by generalized coordinates  $\mathbf{q}$ , the dynamics of *TowrISIR* can be represented in the joint space as follows,

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}, \quad (2.25)$$

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 chapter, all wheels are maintained in contact with the ground.  $\mathbf{J}_c \in \mathbb{R}^{3n_c \times n}$  is the contact Jacobian combining  $n_c$  contact points.  $\mathbf{F}_c \in \mathbb{R}^{3n_c}$  represents the stack of all contact forces.  $\boldsymbol{\tau} \in \mathbb{R}^{n_a}$  denotes the actuated torque and  $\mathbf{S} \in \mathbb{R}^{n_a \times n}$  is one selection matrix of actuated DOFs that are in joints,  $n_a = n_r + n_w$  is the actuated joint number.

The floating-base robot can be divided into under-actuated, fully actuated and over-constrained systems depending on the kinematics-constraint number. The unconstrained system lies in a manifold with  $n - 3n_c$  dimensions while the actuated joints embody in  $n_a$  dimension. When  $3n_c < 6$ , less than two legs touching the ground, the system is under-actuated. When  $3n_c = 6$ , two legs touch the ground, the system becomes fully actuated. Exactly one solution existed in the inverse dynamics problem. When  $3n_c > 6$ , more than two legs touch the ground, the system is over-constrained. There are infinite solutions for actuated  $\boldsymbol{\tau}$  that is able to achieve the desired joint acceleration.

For the derivation of the dynamics model, instead of the conventional notation, the spatial vectors are used to get more compact notations of the equations of the motion (EOM). According to *d'Alembert's Principle*, the virtual work integrating all bodies of the system is given as below,

$$0 = \delta W = \sum_{i=1}^{N_B} [\mathbf{v}_i^T (\dot{\mathbf{h}}_i - {}^i \mathbf{X}_{g_i}^{-T} \mathbf{F}_{g_i} - \mathbf{f}_i + \sum_{j \in c(i)} {}^i \mathbf{X}_j^{-T} \mathbf{f}_j - {}^i \mathbf{X}_{c_i}^{-T} \mathbf{F}_{c_i})], \quad (2.26a)$$

$$\mathbf{0} = \sum_{i=1}^{N_B} [\mathbf{J}_i^T (\dot{\mathbf{h}}_i - {}^i \mathbf{X}_{g_i}^{-T} \mathbf{F}_{g_i} - \mathbf{f}_i + \sum_{j \in c(i)} {}^i \mathbf{X}_j^{-T} \mathbf{f}_j - {}^i \mathbf{X}_{c_i}^{-T} \mathbf{F}_{c_i})]. \quad (2.26b)$$

where the gravity force  $\mathbf{F}_{g_i}$  of body  $i$  is expressed in inertial frame.  $\dot{\mathbf{h}}_i$  can be written by using joint velocity and acceleration as follows,

$$\dot{\mathbf{h}}_i = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i = \mathbf{I}_i \mathbf{J}_i \ddot{\mathbf{q}} + \mathbf{I}_i \dot{\mathbf{J}}_i \dot{\mathbf{q}} + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i. \quad (2.27)$$

For each  $i$ , note that components  $\dot{\mathbf{h}}_i$  and  $\mathbf{f}_i$  are expressed in  $i$ 's local coordinate. The force components  $\mathbf{F}_{g_i}$ ,  $\mathbf{F}_{c_i}$  and  $\mathbf{f}_j$  should also be transferred to act on the origin of the  $i$ 's local frame by  ${}^i \mathbf{X}_{g_i}^{-T}$ ,  ${}^i \mathbf{X}_{c_i}^{-T}$  and

${}^i\mathbf{X}_j^{-T}$ . The actuated torque  $\boldsymbol{\tau}$  has the following relationship,

$$-\mathbf{S}^T \boldsymbol{\tau} = \sum_{i=1}^{N_B} [\mathbf{J}_i^T (-\mathbf{f}_i + \sum_{j \in c(i)} {}^i\mathbf{X}_j^F \mathbf{f}_j)]. \quad (2.28)$$

Then substitute  $\dot{\mathbf{h}}_i$  into (2.26), the elements of EOM can be achieved as below,

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^{N_B} \mathbf{J}_i^T \mathbf{I}_i \mathbf{J}_i, \quad (2.29a)$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^{N_B} [\mathbf{J}_i^T \mathbf{I}_i \dot{\mathbf{J}}_i \dot{\mathbf{q}} + \mathbf{J}_i^T \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i], \quad (2.29b)$$

$$\mathbf{G}(\mathbf{q}) = \sum_{i=1}^{N_B} -\mathbf{J}_i^T {}^i\mathbf{X}_{g_i}^{-T} \mathbf{F}_{g_i} = \sum_{i=1}^{N_B} -\mathbf{J}_{g_i}^T \mathbf{F}_{g_i}, \quad (2.29c)$$

$$\mathbf{S}^T \boldsymbol{\tau} = \sum_{i=1}^{N_B} [-\mathbf{J}_i^T (-\mathbf{f}_i + \sum_{j \in c(i)} {}^i\mathbf{X}_j^F \mathbf{f}_j)], \quad (2.29d)$$

$$\mathbf{J}_c^T \mathbf{F}_c = \sum_{i=1}^{N_B} \mathbf{J}_i^T {}^i\mathbf{X}_{c_i}^{-T} \mathbf{F}_{c_i} = \sum_{i=1}^{N_B} \mathbf{J}_{c_i}^T \mathbf{F}_{c_i}. \quad (2.29e)$$

For real-time control with high frequency,  $\mathbf{M}(\mathbf{q})$  will be calculated using the composite-rigid-body algorithm (CRBA) [Walker 1982].  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  and  $\mathbf{G}(\mathbf{q})$  can be calculated by the recursive Newton-Euler algorithm (RNEA) [Luh 1980], which are modified by [Featherstone 2014] but they are equivalent.

### 2.1.5 Floating-base Centroidal Dynamics

In this paper, we derive two components of centroidal dynamics in a straight way, which is more apparent by comparing with the generalized dynamics model in (2.25). Firstly, the spatial whole-body centroidal momentum is reviewed as follows,

$$\mathbf{h}_G = \begin{bmatrix} \mathbf{k}_G \\ \mathbf{l}_G \end{bmatrix} = \sum_{i=1}^{N_B} {}^i\mathbf{X}_G^T \mathbf{h}_i = \sum_{i=1}^{N_B} {}^i\mathbf{X}_G^T \mathbf{I}_i \mathbf{v}_i = \sum_{i=1}^{N_B} {}^i\mathbf{X}_G^T \mathbf{I}_i \mathbf{J}_i \dot{\mathbf{q}} = \mathbf{X}_G^T \mathbf{I} \mathbf{J} \dot{\mathbf{q}} = \mathbf{A}_G \dot{\mathbf{q}}, \quad (2.30)$$

where  $\mathbf{k}_G$  and  $\mathbf{l}_G$  represent the 3-DOF centroidal angular momentum and translational momentum respectively.  $\mathbf{A}_G \in \mathbb{R}^{6 \times n}$  is the centroidal momentum matrix (CMM). It is noticed that  $\mathbf{A}_G$  and  $\mathbf{M}$  are both composed by spatial inertias of all bodies, and they are expressed as below,

$$\begin{aligned} \mathbf{A}_G &= \sum_{i=1}^{N_B} \mathbf{A}_{G_i} = \sum_{i=1}^{N_B} {}^i\mathbf{X}_G^T \mathbf{I}_i \mathbf{J}_i = \sum_{i=1}^{N_B} {}^b\mathbf{X}_G^T ({}^i\mathbf{X}_b^T \mathbf{I}_i \mathbf{J}_i), \\ \mathbf{M} &= \sum_{i=1}^{N_B} \mathbf{M}_i = \sum_{i=1}^{N_B} \mathbf{J}_i^T \mathbf{I}_i \mathbf{J}_i, \end{aligned} \quad (2.31)$$

where  ${}^b\mathbf{X}_G \in \mathbb{R}^{6 \times 6}$  is the transformation matrix for spatial vectors from whole-body CoM frame to the base frame whose index is 1 or  $b$ .  ${}^i\mathbf{X}_b$  transforms vectors from the base frame to the body frame  $i$ . No matter the index  $i$  is, its Jacobian can always be written with the first 6 columns equal to  ${}^i\mathbf{X}_b \boldsymbol{\Phi}_b = {}^i\mathbf{X}_b$  due to that  $\boldsymbol{\Phi}_b$  is the identity matrix  $\mathbf{1}_{6 \times 6}$ . Thus, one relationship can be derived as follows,

$${}^i\mathbf{X}_1^T = \mathbf{U} \mathbf{J}_i^T, \quad (2.32)$$

$$(2.32) \Rightarrow \mathbf{A}_{G_i} = {}^b\mathbf{X}_G^T ({}^i\mathbf{X}_b^T \mathbf{I}_i \mathbf{J}_i) = {}^b\mathbf{X}_G^T \mathbf{U} \mathbf{M}_i, \quad (2.33)$$

where  $\mathbf{U}$  is a selection matrix  $[\mathbf{1}_{6 \times 6} \ \mathbf{0}_{6 \times (n-6)}]$ . Combine all  $\mathbf{A}_{G_i}$  and  $\mathbf{M}_i, i \in \{1, \dots, N_B\}$ , the centroidal momentum matrix  $\mathbf{A}_G$  can be derived with simple formulation from  $\mathbf{M}$ ,

$$\mathbf{A}_G = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M}, \quad (2.34)$$

where  ${}^1\mathbf{X}_G^T$  is achieved by setting the robot CoM frame to be parallel with the axes of the inertial frame, as follows,

$${}^1\mathbf{X}_G = \begin{bmatrix} {}^1\mathbf{R}_G & \mathbf{0} \\ \mathbf{S}({}^1\mathbf{p}_G) {}^1\mathbf{R}_G & {}^1\mathbf{R}_G \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{R}_0 & \mathbf{0} \\ \mathbf{S}({}^1\mathbf{p}_G) {}^1\mathbf{R}_0 & {}^1\mathbf{R}_0 \end{bmatrix}, \quad (2.35)$$

where  ${}^1\mathbf{p}_G = {}^b\mathbf{p}_G$  represents the robot-CoM position in the base frame which can be extracted from well-structured  $\mathbf{I}_1^C$  or  $\mathbf{I}_b^C$  [Siciliano 2016] as below,

$$\mathbf{I}_1^C = \mathbf{I}_b^C = \begin{bmatrix} \bar{\mathbf{I}}_1^C & m_G \mathbf{S}({}^1\mathbf{p}_G) \\ m_G \mathbf{S}({}^1\mathbf{p}_G)^T & m_G \mathbf{1} \end{bmatrix}, \quad (2.36)$$

$${}^1\mathbf{p}_G = \frac{1}{m_G} \left[ \mathbf{I}_{1(3,5)}^C \ \mathbf{I}_{1(1,6)}^C \ \mathbf{I}_{1(2,4)}^C \right]^T. \quad (2.37)$$

where  $m_G$  is the robot mass. Then the whole-body centroidal dynamics is reviewed which is the derivative of (2.30) as follows,

$$\dot{\mathbf{h}}_G = \begin{bmatrix} \dot{\mathbf{k}}_G \\ \dot{\mathbf{l}}_G \end{bmatrix} = \dot{\mathbf{A}}_G \dot{\mathbf{q}} + \mathbf{A}_G \ddot{\mathbf{q}}, \quad (2.38)$$

where  $\dot{\mathbf{A}}_G \dot{\mathbf{q}}$  is the centroidal momentum bias force (CMBF). It can also be derived straightly by comparing with the Centripetal and Coriolis force as below,

$$\dot{\mathbf{A}}_G \dot{\mathbf{q}} = \sum_{i=1}^{N_B} (\dot{\mathbf{A}}_G \dot{\mathbf{q}})_i = \sum_{i=1}^{N_B} {}^b\mathbf{X}_G^T ({}^i\mathbf{X}_b^T \mathbf{I}_i \dot{\mathbf{J}}_i \dot{\mathbf{q}} + {}^i\mathbf{X}_b^T \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i), \quad (2.39)$$

$$\mathbf{C} = \sum_{i=1}^{N_B} \mathbf{C}_i = \sum_{i=1}^{N_B} \{ \mathbf{J}_i^T \mathbf{I}_i \dot{\mathbf{J}}_i \dot{\mathbf{q}} + \mathbf{J}_i^T \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i \}, \quad (2.40)$$

therefore, CMBF can be derived using (2.32) as follows,

$$(\dot{\mathbf{A}}_G \dot{\mathbf{q}})_i = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{C}_i \Rightarrow \dot{\mathbf{A}}_G \dot{\mathbf{q}} = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{C}. \quad (2.41)$$

The operational-space gravity force at the robot CoM  $\boldsymbol{\rho}_G$  can be derived with the same comparison method as below,

$$\boldsymbol{\rho}_G = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{G}. \quad (2.42)$$

Then the operational space force at the robot CoM  $\mathbf{F}_G$  can be expressed as follows,

$$\mathbf{F}_G = {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M} \ddot{\mathbf{q}} + {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{C} + {}^1\mathbf{X}_G^T \mathbf{U} \mathbf{G} = {}^1\mathbf{X}_G^T \mathbf{U} (\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}). \quad (2.43)$$

When the generalized dynamics model is obtained, the above relationship represents a quite efficient formulation to calculate the centroidal dynamics equation and its relating operational space task force.

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 [Wensing 2016],

$$\mathbf{J}_G = \mathbf{I}_G^{-1} ({}^b\mathbf{X}_G^T \mathbf{U} \mathbf{M}) = ({}^b\mathbf{X}_G)^{-1} (\mathbf{I}_b^C)^{-1} \mathbf{U} \mathbf{M}, \quad (2.44)$$

where  $\mathbf{I}_G = {}^b\mathbf{X}_G^T \mathbf{I}_b^C {}^b\mathbf{X}_G$  denotes the composite rigid-body inertia at the robot CoM, referred in [Siciliano 2016].  $\mathbf{I}_b^C = \mathbf{U} \mathbf{M} \mathbf{U}^T$  is the base composite inertia and  $\mathbf{U} = [\mathbf{1}_{6 \times 6} \ \mathbf{0}_{6 \times (n-6)}]$  is a selection matrix. The robot CoM frame is shown in Fig. 2.2.  ${}^b\mathbf{X}_G \in \mathbb{R}^{6 \times 6}$  is the transformation matrix

for spatial vectors from the robot CoM frame to the base frame whose index is 1 or  $b$ .  $\mathbf{J}_G$  can also be decomposed as below,

$$\mathbf{J}_G = [\mathbf{J}_G^b \quad \mathbf{J}_G^g] = [\mathbf{J}_G^b \quad \mathbf{J}_G^{r1} \quad \mathbf{J}_G^{w1} \quad \mathbf{J}_G^{r2} \quad \mathbf{J}_G^{w2} \quad \cdots \quad \mathbf{J}_G^{r4} \quad \mathbf{J}_G^{w4}], \quad (2.45)$$

where  $\mathbf{J}_G^b \in \mathbb{R}^{6 \times 6}$  and  $\mathbf{J}_G^g \in \mathbb{R}^{6 \times (n-6)}$  relate to the floating-base motion, and the leg  $i$  joint motion respectively.  $\mathbf{J}_G^{r_i} \in \mathbb{R}^{6 \times 3}$  corresponds to the leg  $i$  revolute-joint motion, and  $\mathbf{J}_G^{w_i} \in \mathbb{R}^{6 \times 1}$  relates to the leg  $i$  wheel-joint motion. They will be used to decompose the centroidal momentum model in Chapter 4. Another general formulation to compute the CMM and CMBF is given by the OSC method, and these two methods are equivalent,

$$\mathbf{F}_G = \bar{\mathbf{J}}_G^T (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}) = \mathbf{\Lambda}_G \ddot{\mathbf{x}}_G + \boldsymbol{\mu}_G + \boldsymbol{\rho}_G. \quad (2.46)$$

where  $\mathbf{\Lambda}_G$ ,  $\boldsymbol{\mu}_G$  and  $\boldsymbol{\rho}_G$  denote the operational space inertia, Centripetal and Coriolis force, and gravity force at the robot CoM.  $\bar{\mathbf{J}}_G$  is the dynamically-consistent inverse of the Jacobian  $\mathbf{J}_G$ , detailed in the following section. Therefore, CMM and CMBF can be derived as follows,

$$\mathbf{A}_G = \bar{\mathbf{J}}_G^T \mathbf{M} = (\mathbf{J}_G \mathbf{M}^{-1} \mathbf{J}_G^T)^{-1} \mathbf{J}_G, \quad (2.47a)$$

$$\dot{\mathbf{A}}_G \dot{\mathbf{q}} = \bar{\mathbf{J}}_G^T \mathbf{C} = (\mathbf{J}_G \mathbf{M}^{-1} \mathbf{J}_G^T)^{-1} \mathbf{J}_G \mathbf{M}^{-1} \mathbf{C}. \quad (2.47b)$$

## 2.2 Operational Space Torque Control

In this section, we present the general control framework. Then we review the dynamically-consistent pseudo-inverse based operational space control (OSC). Three kinds of projections relating to the contact constraint are reviewed, which enable the generalized dynamics model to be independent of the contact forces. Finally, we review the prioritized OSC and we propose one more compact formulations in a recursively, which serve as the basis for the development and improvement of our torque controllers in next chapter.

### 2.2.1 General Control Framework

The general control framework is shown in Fig. 2.4, which contains the motion reference, centroidal dynamics based motion generator, model based torque controller and the simulated robot. By given the high-level trajectory references relating to the base or the robot centroidal motion, this thesis concentrates on generating the robot whole-body motion by taking advantages of the robot centroidal properties and by improving the torque controllers in the state of the art. In each control loop, various sensors on the robot send feedback information of the robot whole-body motion states to other three parts, which close the control loop.

This control framework is common for the quadruped-on-wheel robot, *TowrISIR*, and for the quadruped robot, *QuadISIR*. The centroidal properties are applied in the developed motion generators for both two robots, detailed in Chapter 4 and Chapter 5 respectively. The improved model based torque controllers in Chapter 3 are also verified on our two robots, which can also be used on other high-DOF robots, e.g. humanoids.

### 2.2.2 Operational Space Control (OSC)

In this section, a brief introduction is about torque-based operational space control [Khatib 1987, Sentis 2007]. The standard dynamics model for a operational space task  $\dot{\mathbf{x}}_t = \mathbf{J}_t(\mathbf{q})\dot{\mathbf{q}}$  can be expressed as below,

$$\mathbf{\Lambda}_t(\mathbf{q})\ddot{\mathbf{x}}_t + \boldsymbol{\mu}_t(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\rho}_t(\mathbf{q}) = \bar{\mathbf{J}}_t^T \mathbf{S}^T \boldsymbol{\tau} + \bar{\mathbf{J}}_t^T \mathbf{J}_c^T \mathbf{F}_c \quad (2.48)$$

where the subscript  $t$  denotes the task index.  $\mathbf{\Lambda}_t(\mathbf{q})$  denotes the operational space inertia matrix,  $\boldsymbol{\mu}_t(\mathbf{q}, \dot{\mathbf{q}})$  represents the Centripetal and Coriolis force in task space and  $\boldsymbol{\rho}_t(\mathbf{q})$  is the gravity-dependent force. The

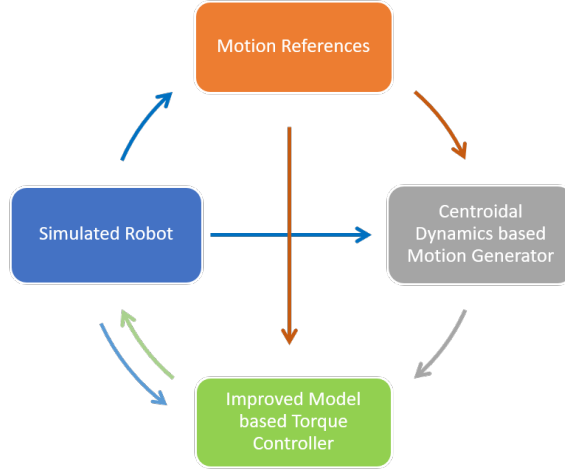


Figure 2.4: Whole-Body Dynamics Control Scheme of TowrISIR

above model is obtained by multiplying  $\bar{\mathbf{J}}_t^T$  at both sides of the generalized dynamics model in (2.25), and  $\bar{\mathbf{J}}_t$  is the dynamically-consistent generalized inverse of the task Jacobian  $\mathbf{J}_t$ . These quantities are given as follows,

$$\Lambda_t(\mathbf{q}) = (\mathbf{J}_t \mathbf{M}^{-1} \bar{\mathbf{J}}_t^T)^{-1}, \quad (2.49a)$$

$$\bar{\mathbf{J}}_t^T(\mathbf{q}) = \Lambda_t \mathbf{J}_t \mathbf{M}^{-1}, \quad (2.49b)$$

$$\boldsymbol{\mu}_t(\mathbf{q}, \dot{\mathbf{q}}) = \bar{\mathbf{J}}_t^T \mathbf{C} - \Lambda_t \dot{\mathbf{J}}_t \dot{\mathbf{q}}, \quad (2.49c)$$

$$\boldsymbol{\rho}_t(\mathbf{q}) = \bar{\mathbf{J}}_t^T \mathbf{G}. \quad (2.49d)$$

When the operational space task is established at the contact point, the EOM in contact task space can be derived and the contact force  $\mathbf{F}_c$  can be expressed as below,

$$\Lambda_c(\mathbf{q}) \ddot{\mathbf{x}}_c + \boldsymbol{\mu}_c(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\rho}_c(\mathbf{q}) = \bar{\mathbf{J}}_c^T \mathbf{S}^T \boldsymbol{\tau} + \mathbf{F}_c. \quad (2.50)$$

Due to the contact constraints  $\ddot{\mathbf{x}}_c = \mathbf{0}$ , the contact force can be achieved as follows,

$$\mathbf{F}_c = \boldsymbol{\mu}_c(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\rho}_c(\mathbf{q}) - \bar{\mathbf{J}}_c^T \mathbf{S}^T \boldsymbol{\tau}. \quad (2.51)$$

Substitute the contact force in (2.51) into operational space dynamics in (2.48), we can obtain the contact-consistent dynamics model for one task in the operational space [Park 2006] as follows,

$$\tilde{\Lambda}_t(\mathbf{q}) \ddot{\mathbf{x}}_t + \tilde{\boldsymbol{\mu}}_t(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\boldsymbol{\rho}}_t(\mathbf{q}) = \tilde{\bar{\mathbf{J}}}_t^T \mathbf{S}^T \boldsymbol{\tau}, \quad (2.52)$$

where the quantities can be expressed as below,

$$\tilde{\Lambda}_t(\mathbf{q}) = [\mathbf{J}_t \mathbf{M}^{-1} (\mathbf{1} - \mathbf{J}_c^T \bar{\mathbf{J}}_c^T) \bar{\mathbf{J}}_t^T]^{-1}, \quad (2.53a)$$

$$\tilde{\bar{\mathbf{J}}}_t^T(\mathbf{q}) = \tilde{\Lambda}_t \mathbf{J}_t \mathbf{M}^{-1} (\mathbf{1} - \mathbf{J}_c^T \bar{\mathbf{J}}_c^T), \quad (2.53b)$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{q}, \dot{\mathbf{q}}) = \tilde{\bar{\mathbf{J}}}_t^T \mathbf{C} + \tilde{\Lambda}_t \mathbf{J}_t \mathbf{M}^{-1} \bar{\mathbf{J}}_c^T \Lambda_c \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \tilde{\Lambda}_t \dot{\mathbf{J}}_t \dot{\mathbf{q}}, \quad (2.53c)$$

$$\tilde{\boldsymbol{\rho}}_t(\mathbf{q}) = \tilde{\bar{\mathbf{J}}}_t^T \mathbf{G}. \quad (2.53d)$$

Another derivation method for the contact-constraint consistent dynamics will be described in the next subsection using prioritized task control.

### 2.2.3 Contact-constraint Consistent Projections

Substitute  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\rho}_c$  in (2.49) into the contact force in (2.51), we can obtain the contact force using the joint-space parameters as follows,

$$\mathbf{F}_c = (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T)^{-1} [\mathbf{J}_c \mathbf{M}^{-1} (\mathbf{C} + \mathbf{G} - \mathbf{S}^T \boldsymbol{\tau}) + \mathbf{J}_c \ddot{\mathbf{q}}], \quad (2.54)$$

This can be substituted back to the generalized dynamics model which is rearranged as below,

$$\mathbf{P} [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}] = \mathbf{P} \mathbf{S}^T \boldsymbol{\tau}. \quad (2.55)$$

This projection process by the linear projector  $\mathbf{P}$  maps the robot dynamics onto the manifold with contact-consistent constraint and enables the EOM independent of the contact force  $\mathbf{F}_c$ . From (2.55), the actuated torque  $\boldsymbol{\tau}$  can be obtained by the Moore-Penrose inverse method as follows,

$$\boldsymbol{\tau} = (\mathbf{P} \mathbf{S}^T)^+ \mathbf{P} (\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}). \quad (2.56)$$

This formulation gives a general projection formulation in which  $\mathbf{P}$  is not unique in the state of the art. In this section, three methods will be introduced.

#### Direct Projector:

The direct projector is derived by combining (2.54), (2.55) and the generalized dynamics model in (2.25), as follows,

$$\mathbf{P}_{direct} = \mathbf{1} - \mathbf{J}_c^T (\mathbf{J}_c \mathbf{M}^{-1} \mathbf{J}_c^T)^{-1} \mathbf{J}_c \mathbf{M}^{-1}. \quad (2.57)$$

In addition, (2.55) can also be derived from the dynamically contact-consistent null-space projector  $\mathbf{N}_c$  [Sentis 2005] which ensures that other tasks do not introduce acceleration components into the contact-constrained directions.  $\mathbf{N}_c$  is expressed as below,

$$\mathbf{N}_c = \mathbf{1} - \bar{\mathbf{J}}_c \mathbf{J}_c, \quad (2.58)$$

We project the joint-space dynamics EOM in (2.25) into the null space of the contact constraints as follows,

$$\mathbf{N}_c^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}] = \mathbf{N}_c^T \mathbf{S}^T \boldsymbol{\tau}. \quad (2.59)$$

Compared (2.57) and (2.58), the direct projector is equal to the transpose of contact-constraint null space as below,

$$\mathbf{P}_{direct} = \mathbf{N}_c^T, \quad (2.60)$$

where the properties  $\mathbf{N}_c^T \mathbf{J}_c^T = \mathbf{0}$ ,  $\mathbf{J}_c \bar{\mathbf{J}}_c \mathbf{J}_c = \mathbf{J}_c$  and (2.49b) are applied.

#### Operational Space Projector:

For the contact-independent dynamics model in (2.59), and the actuated torque formulation can be obtained by applying the dynamically-consistent inverse of  $\mathbf{N}_c^T \mathbf{S}^T$  [Sentis 2010] as follows,

$$\begin{aligned} \boldsymbol{\tau} &= (\mathbf{S} \mathbf{N}_c \mathbf{M}^{-1} (\mathbf{S} \mathbf{N}_c)^T)^+ (\mathbf{S} \mathbf{N}_c) \mathbf{M}^{-1} \mathbf{N}_c^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}], \\ &= (\mathbf{S} \mathbf{N}_c \mathbf{M}^{-1} \mathbf{S}^T)^+ \mathbf{S} \mathbf{N}_c \mathbf{M}^{-1} [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}], \end{aligned} \quad (2.61)$$

where we apply the null-space property as below,

$$\mathbf{N}_c \mathbf{M}^{-1} = \mathbf{M}^{-1} \mathbf{N}_c^T = \mathbf{N}_c \mathbf{M}^{-1} \mathbf{N}_c^T, \quad (2.62)$$

Compare (2.61) with the general formulation in (2.56), the projector in (2.55) can be expressed as follows,

$$\mathbf{P}_{osc} = \mathbf{S} \mathbf{N}_c \mathbf{M}^{-1} = \mathbf{S} \mathbf{M}^{-1} \mathbf{N}_c^T = \mathbf{S} \mathbf{N}_c \mathbf{M}^{-1} \mathbf{N}_c^T. \quad (2.63)$$

#### QR Decomposition Projector:

[Mistry 2010] applied the orthogonal decomposition of the constraint Jacobian and proposed another inverse dynamics control law as follows,

$$\mathbf{J}_c^T = \mathbf{Q} [\mathbf{R}^T \quad \mathbf{0}_{3n_c \times (n-3n_c)}]^T, \quad (2.64)$$

where the orthogonal matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  with property  $\mathbf{Q}^T = \mathbf{Q}^{-1}$ .  $\mathbf{R} \in \mathbb{R}^{3n_c \times 3n_c}$  is an upper right triangular matrix with all positive diagonal elements. Substitute  $\mathbf{J}_c^T$  into the joint dynamics EOM in (2.25),  $\mathbf{Q}^T$  is multiplied on its both sides. Then (2.25) is decomposed by two selection matrices to two independent equations of the system dynamics as below,

$$\mathbf{S}_c \mathbf{Q}^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}] = \mathbf{S}_c \mathbf{Q}^T \mathbf{S}^T \boldsymbol{\tau} + \mathbf{R} \mathbf{F}_c, \quad (2.65a)$$

$$\mathbf{S}_u \mathbf{Q}^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}] = \mathbf{S}_u \mathbf{Q}^T \mathbf{S}^T \boldsymbol{\tau}, \quad (2.65b)$$

where  $\mathbf{S}_c = [\mathbf{1}_{k \times k} \quad \mathbf{0}_{k \times (n-k)}]$  and  $\mathbf{S}_u = [\mathbf{0}_{(n-k) \times k} \quad \mathbf{1}_{(n-k) \times (n-k)}]$ . Then the torque formulation and the contact force are achieved as follows,

$$\boldsymbol{\tau} = (\mathbf{S}_u \mathbf{Q}^T \mathbf{S}^T)^+ \mathbf{S}_u \mathbf{Q}^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}], \quad (2.66a)$$

$$\mathbf{F}_c = \mathbf{R}^{-1} \mathbf{S}_c \mathbf{Q}^T [\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G}]. \quad (2.66b)$$

Therefore, the QR decomposition projector can be written as follows,

$$\mathbf{P}_{QR} = \mathbf{S}_u \mathbf{Q}^T. \quad (2.67)$$

Although different projectors have different formulations with different calculation complexity, the actuated torque remain the same results [Righetti 2011b]. Compared among these three projectors,  $\mathbf{P}_{QR}$  has more strengths in handling model uncertainty, because there exists  $\mathbf{M}^{-1}$  in  $\mathbf{P}_{direct}$  and  $\mathbf{P}_{osc}$  in real-time control whose precision depends on the model establishment, and the calculation complexity increases a lot due to inverse of  $\mathbf{M}$ .

## 2.2.4 Contact Constraint consistent Prioritized OSC

In this section, we will review the contact-constraint consistent prioritized OSC. By given  $N$  prioritized tasks in the operational space, the actuated torque  $\boldsymbol{\tau}$  can be achieved which respects the task priorities and the contact constraints. The general formulation of task  $t$  operational space force  $\mathbf{F}_{t|pre(t)}$  is achieved by given the task motion reference  $\ddot{\mathbf{x}}_{t(ref)}$  as follows,

$$\mathbf{F}_{t|pre(t)} = \boldsymbol{\Lambda}_{t|pre(t)} \ddot{\mathbf{x}}_{t(ref)} + \boldsymbol{\mu}_{t|pre(t)} + \boldsymbol{\rho}_{t|pre(t)} - \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \mathbf{M}^{-1} \sum_{i=1}^{t-1} \mathbf{J}_{i|pre(i)}^T \mathbf{F}_{i|pre(i)}, \quad (2.68)$$

where  $\boldsymbol{\Lambda}_{t|pre(t)}$ ,  $\boldsymbol{\mu}_{t|pre(t)}$  and  $\boldsymbol{\rho}_{t|pre(t)}$  are the extended general inertia, Coriolis/centrifugal and gravity vectors.  $\mathbf{J}_{t|pre(t)} = \mathbf{J}_t \mathbf{N}_{pre(t)}$  denotes the prioritized Jacobian for the operational-space task  $i$ , which are projected into the null-space of the previous tasks, indexed as  $pre(t)$ .  $\mathbf{N}_{pre(t)}$  is the null-space by combining all tasks above level  $i$  which can be calculated recursively as  $\mathbf{N}_{pre(t+1)} = \mathbf{N}_{pre(t)} - \bar{\mathbf{J}}_{t|pre(t)}^T \mathbf{J}_{t|pre(t)}$ . Furthermore,  $\mathbf{N}_{pre(1)} = \mathbf{N}_c$  where  $\mathbf{N}_c$  denotes the null-space projector of the Jacobiana combing all contact points, which means that all designed tasks should satisfy the contact constraints.  $\bar{\mathbf{J}}_{t|pre(t)}$  is the prioritized contact-consistent and dynamics-consistent inverse of the task  $t$  Jacobian. These variables are listed as follows,

$$\boldsymbol{\Lambda}_{t|pre(t)} = (\mathbf{J}_{t|pre(t)} \mathbf{M}^{-1} \bar{\mathbf{J}}_{t|pre(t)}^T)^{-1}, \quad (2.69a)$$

$$\mathbf{J}_{t|pre(t)} = \mathbf{J}_t \mathbf{N}_{t|pre(t)}, \quad (2.69b)$$

$$\bar{\mathbf{J}}_{t|pre(t)}^T = \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_{t|pre(t)} \mathbf{M}^{-1}, \quad (2.69c)$$

$$\mathbf{N}_{pre(t)} = \prod_{i=1}^{t-1} \mathbf{N}_{i|pre(i)} = \mathbf{N}_c - \sum_{i=0}^{t-1} \bar{\mathbf{J}}_{i|pre(i)}^T \mathbf{J}_{i|pre(i)}, \quad (2.69d)$$

$$\boldsymbol{\mu}_{t|pre(t)} = \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{C} + \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \bar{\mathbf{J}}_c \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \boldsymbol{\Lambda}_{t|pre(t)} \dot{\mathbf{J}}_t \dot{\mathbf{q}}, \quad (2.69e)$$

$$\boldsymbol{\rho}_{t|pre(t)} = \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \mathbf{M}^{-1} \mathbf{N}_c^T \mathbf{G}. \quad (2.69f)$$

The contact-constraint consistent actuated torque is derived as below,

$$\boldsymbol{\tau} = (\mathbf{P}\mathbf{S}^T)^+ \mathbf{P} \sum_{t=1}^N [\mathbf{J}_{t|pre(t)}^T \mathbf{F}_{t|pre(t)}], \quad (2.70)$$

where  $\mathbf{P}$  equals to  $\mathbf{P}_{direct}$ ,  $\mathbf{P}_{osc}$  or  $\mathbf{P}_{QR}$ . The detailed null-space based inverse dynamics is referred to [Sentis 2005].

For optimization based techniques, each task acceleration  $\ddot{\mathbf{x}}_t$  should be expressed by the actuated torque. From (2.68),  $\ddot{\mathbf{x}}_t$  can be obtained with the quadratic form as follows,

$$\ddot{\mathbf{x}}_t = \mathbf{A}_t^T \boldsymbol{\tau} + \mathbf{B}_t, \quad (2.71)$$

where  $\ddot{\mathbf{x}}_t$  depends on previous  $t - 1$  tasks and needs careful derivations, which is not suitable for fast calculation. Therefore, we propose the following formulas which can be calculated in a recursive way, as below,

$$\mathbf{A}_t^T = \boldsymbol{\Lambda}_{t|pre(t)}^{-1} [\mathbf{A}_{F_t}^T + \mathbf{A}_{T_t}^T], \quad (2.72a)$$

$$\mathbf{B}_t = \boldsymbol{\Lambda}_{t|pre(t)}^{-1} [\mathbf{B}_{F_t} + \mathbf{B}_{T_t} - \boldsymbol{\mu}_{t|pre(t)} - \boldsymbol{\rho}_{t|pre(t)}], \quad (2.72b)$$

where  $\mathbf{A}_t$  and  $\mathbf{B}_t$  can be derived recursively using formulas as follows,

$$\mathbf{A}_{F_t}^T = \bar{\mathbf{J}}_{t|pre(t)}^T (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T), \quad (2.73a)$$

$$\mathbf{B}_{F_t} = \bar{\mathbf{J}}_{t|pre(t)}^T \mathbf{J}_c^T \mathbf{B}_c, \quad (2.73b)$$

$$\mathbf{A}_{s_t}^T = \mathbf{A}_{s_{p(t)}}^T + \mathbf{J}_{p(t)|pre(p(t))}^T \mathbf{A}_{F_{p(t)}}^T, \quad (2.73c)$$

$$\mathbf{B}_{s_t} = \mathbf{B}_{s_{p(t)}} + \mathbf{J}_{p(t)|pre(p(t))}^T \mathbf{B}_{F_{p(t)}}, \quad (2.73d)$$

$$\mathbf{A}_{T_t}^T = \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \mathbf{M}^{-1} \mathbf{A}_{s_t}^T, \quad (2.73e)$$

$$\mathbf{B}_{T_t} = \boldsymbol{\Lambda}_{t|pre(t)} \mathbf{J}_t \mathbf{M}^{-1} \mathbf{B}_{s_t}, \quad (2.73f)$$

$$\mathbf{A}_c^T = -\bar{\mathbf{J}}_c^T \mathbf{S}^T, \quad (2.73g)$$

$$\mathbf{B}_c = \boldsymbol{\mu}_c + \boldsymbol{\rho}_c, \quad (2.73h)$$

where  $p(t)$  is the processor task priority of task  $t$ , namely  $p(t) = t - 1$ .  $\mathbf{A}_{F_t}$ ,  $\mathbf{B}_{F_t}$ ,  $\mathbf{A}_{T_t}$  and  $\mathbf{B}_{T_t}$  are middle-process variables.  $\mathbf{A}_{s_t}$  and  $\mathbf{B}_{s_t}$  are used to allow  $\mathbf{A}_{T_t}$  and  $\mathbf{B}_{T_t}$  to be calculated in a recursive way. These proposed formulas in (2.72) and (2.73) can also be referred in our published weighted torque control [Du 2020a], which is improved to be more compact in the next chapter.

## 2.3 Summary

This chapter describes the whole-body kinematics and dynamics of our quadruped-on-wheel robot *TowrISIR*, in which all the formulations are derived by spatial notations. The inverse kinematics model is decomposed into several components relating to the base motion, the legged motion and the wheel motion, which serves as the basis for the model based wheeled motion in Chapter 4. The contact point is estimated at the lowest points of the wheels. We derive the centroidal dynamics in a more apparent way and the centroidal properties will be applied in the following chapters. Then we propose the general whole-body control framework which is common for two four-limb robots in this thesis. Furthermore, we review the operational space control and three contact-constraint consistent projectors. Finally, we introduce the multi-task prioritized OSC and then propose one set of recursive formulas to extract the quadratic form of each task acceleration. The contents in this chapter function as basis for the developed motion generators and improved torque controllers in following chapters.



# Prioritized Impedance Force based Hierarchical Torque Control

---

## Contents

<b>3.1</b>	<b>Introduction and Chapter Outline</b>	<b>37</b>
<b>3.2</b>	<b>Generalized Dynamics Model for a Locomotion Robot</b>	<b>37</b>
<b>3.3</b>	<b>Whole-body Impedance Controllers</b>	<b>38</b>
<b>3.4</b>	<b>Prioritized Impedance Controller based Torque Control</b>	<b>40</b>
3.4.1	Modified Dynamics Model for Pseudo-inverse based Dynamics	40
3.4.2	Prioritized Impedance Controller based Null-space based Prioritized Weighted Optimization (P-WQP)	41
3.4.3	One New Dynamics Model for Prioritized Impedance Controller based Hierarchical Optimization (P-HQP)	42
3.4.4	Integration of the Prioritized Weighted Scheme in the Hierarchical Optimization (P-HWQP)	47
<b>3.5</b>	<b>Simulation</b>	<b>48</b>
3.5.1	Simulations for the Prioritized Weighted Optimization	50
3.5.2	Simulations for the Hybrid Hierarchical-weighted Optimization	52
<b>3.6</b>	<b>Summary</b>	<b>55</b>

---

This chapter proposes the concept called prioritized impedance controller (PIC). We integrate it into four kinds of improved hierarchical operational-space torque control frameworks: one pseudo-inverse based dynamics method and three optimization-based techniques, including one null-space based weighted optimization solver, one hierarchical optimization solver, and one hybrid hierarchical-weighted solver. The integrated PIC enables various impedance control forces to be consistent and compatible with multi-task hierarchies. The concept can handle more general impedance controllers, including the ones built directly on specific designed tasks, and ones indirectly acting on the whole-body behaviors.

For the pseudo-inverse based dynamics method and the null-space based weighted scheme, each task acceleration with the prioritized impedance controller is extracted using one proposed modified dynamics model with PIC. In contrast with the conventional weighted QP, we firstly propose that priorities exist in the weighted scheme, and the proposed model enables to integrate this new weighted scheme into the hierarchical control framework. For the hierarchical optimization solver, another proposed new dynamics model with PIC is proposed to embody the multi-level control hierarchy, and it is used to extract each task acceleration with the quadratic form. The new dynamics model enables the achievement of the more general and complete hierarchical optimization solver, both for the task acceleration extraction and the same number of constraints, and enables various impedance controllers to be integrated compatibly with priorities. The inequality constraints are modified to be compatible with the optimization solver by extracting the quadratic forms of the contact force and the joint acceleration from our new dynamics model. Then we use our proposed model for a more complex scenario which combines the hierarchical and the null-space based weighted schemes together. The detailed algorithm is developed which is more general for the pure weighted scheme, or the pure hierarchical scheme, or the hybrid combination. Each of the approaches can be selected depending on the actual application. Our methods are tested in several

simulation scenarios using a virtual quadruped-on-wheel robot, Towr*ISIR* with one manipulator on its floating base.

### 3.1 Introduction and Chapter Outline

According to the state of the art relating to the multi-task active compliance and the optimization based torque control in Chapter Introduction, we have the following opinions. For the conventional weighted convex optimization without null space projections, when there are not enough DOFs for the robot to finish all tasks feasibly, unfeasible motion directions cannot be removed from the tasks. Furthermore, conventional feedback based controllers can influence all tasks together which cannot be accepted. These two problems can be avoided in this chapter by using null space projections, which is always applied in the hierarchical optimization framework, but neglected in the weighted theme. For the strict hierarchical optimization methods in the state of the art, the strict hierarchy is always achieved by defining specific equality and inequality constraints. Although null-space projections are applied in few works to decouple the influences among hierarchies, as the hierarchy index increases, the number of the constraints also increases, especially for the inequality constraints relating to the friction cone and the joint configurations. The method is not general because the generalized dynamics model is applied which do not embody the multi-level control hierarchy. The existing hybrid hierarchical-weighted schemes also have the above issues. In addition, in the state of the art, for each task reference, the specific feedback impedance controller can be added to enable the compliance on the specific task. However, the general impedance controllers which indirectly influence the whole-body motions are not prioritized which influence all designed tasks together.

In this chapter, we will handle these issues and make the hierarchical compliant control frameworks to be more general and complete. Multiple active compliant controllers are proposed to be integrated into the prioritized operational space control framework, which means that the impedance controllers are consistent with the task hierarchies. In other words, the impedance controllers should influence the designed tasks in a decoupled way. For the pseudo-inverse based dynamics method and null-space based weighted optimization solver, we propose to modify the generalized dynamics model to contain the prioritized impedance controller. In contrast with the conventional weighted QP, we firstly propose that priorities exist in the weighted scheme, and the proposed model enables to integrate this new weighted scheme into the hierarchical control framework. For the optimization based hierarchical techniques, including strict hierarchical convex optimization and a hybrid hierarchical-weighted scheme, we propose another new dynamics model which is more general for hierarchical optimization. The modified or new dynamics model enables the achievement of the more general and complete hierarchical optimization solvers, both for the task acceleration extraction and the same number of constraints. We implement these control schemes on Towr*ISIR* to test the usefulness and validity of our approaches.

The remainder of this chapter is structured as follows: In Section 3.2, system modeling is firstly introduced. The whole-body impedance controllers are introduced in Section 3.3. Then we show how to integrate our prioritized impedance controller in the pseudo-inverse dynamics method, in the null-space based weighted QP, in the hierarchical convex optimization and in hybrid hierarchical-weighted control scheme in Section 3.4.1, Section 3.4.2, Section 3.4.3 and Section 3.4.4, respectively. Then in Section 3.5, we test our method through numerical simulations with our virtual quadruped-on-wheel robot. Section 3.6 closes the chapter with the conclusion.

### 3.2 Generalized Dynamics Model for a Locomotion Robot

A legged robot has  $n$  DOF in which 6 DOFs are relating to the floating base motion. The Towr*ISIR* robot is applied in this chapter and one manipulator is installed on its floating base, shown in Fig. 3.1. The joint connectivity of quadruped-on-wheel manipulator is depicted in Fig. 3.2. The dynamics model has the same formulation as (2.25), which is re-written as follows,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}, \quad (3.1)$$

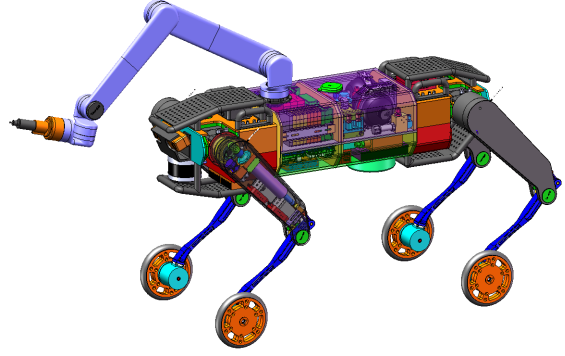
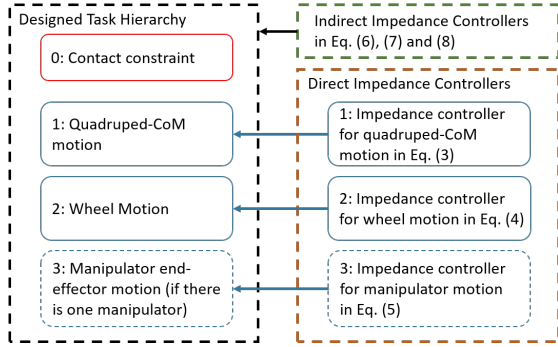


Figure 3.1: [Left]: Task hierarchy definition. The smaller index, the higher priority. Index 0 denotes the contact constraint. For each priority, one specific impedance controller is applied. Other indirect impedance controllers can be integrated to be consistent with the task hierarchy using our approach in this chapter. [Right]: TowrISIR is equipped with one manipulator on its floating base to enable the robot with manipulation behaviors. The manipulator has 6 DOFs and its mass is about  $7kg$ . The weight of the robot with the manipulator is about  $107kg$ .

where  $\mathbf{q} = [\boldsymbol{\theta}_b^T \ \mathbf{p}_b^T \ \mathbf{q}_g^T \ \mathbf{q}_m^T]^T$  represents the joint configuration of the quadruped-on-wheel manipulator.  $\mathbf{q}_g$  is defined in Section 2.1.1 which includes all leg and wheel joints.  $\mathbf{q}_m$  denotes the manipulator joint configuration. We define  $\mathbf{q}_s = [\mathbf{q}_g^T \ \mathbf{q}_m^T]^T$  which is used in the establishment of the inequality constraints.  $\boldsymbol{\tau} \in \mathbb{R}^{n_a}$  is the torque for actuators in each leg and the manipulator.  $\mathbf{S} \in \mathbb{R}^{n_a \times n}$  is the relative selection matrix.  $n_a = 22$  and  $n = 28$  for the quadruped-on-wheel manipulator. In this chapter, we use the rolling-mode locomotion, which means that four legs are maintained in contact with the ground,  $\mathbf{J}_c \in \mathbb{R}^{3n_c \times n}$  and  $n_c = 4$ . Assume that there is no slippage between the contact points and the ground, the velocity of all contact points and their time derivative share the formulations in (2.17).

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

$$\mathbf{J}_G = [{}^1\mathbf{X}_G^T (\mathbf{U} \mathbf{M}_{quad} \mathbf{U}^T) {}^1\mathbf{X}_G]^{-1} ({}^1\mathbf{X}_G^T \mathbf{U} \mathbf{M}_{quad}), \quad (3.2)$$

$$\mathbf{M}_{quad} = \sum \mathbf{J}_i^T \mathbf{I}_i \mathbf{J}_i, \quad i \in \text{Quadruped Bodies} \quad (3.3)$$

where  $\mathbf{U} = [\mathbf{1}_{6 \times 6}, \mathbf{0}_{6 \times (n-6)}]$ . It is noticed that  $\mathbf{M}_{quad} \in \mathbb{R}^{n \times n}$  is the quadruped mass matrix, which means that components in  $\mathbf{M}_{quad}$  relating to the manipulator bodies are equal to zero. And  $\mathbf{M}_{quad}$  is one by-product when getting the whole-body generalized mass matrix  $\mathbf{M}$  through the Composite Rigid Body algorithm [Walker 1982]. The quadruped composite inertia in the base frame is  $\mathbf{I}_{quad} = \mathbf{U} \mathbf{M}_{quad} \mathbf{U}^T$  and the composite rigid body inertia at the quadruped CoM is  $\mathbf{I}_G = {}^1\mathbf{X}_G^T \mathbf{I}_{quad} {}^1\mathbf{X}_G$ . With  $\mathbf{J}_G$ , tasks relating to quadruped CoM can be integrated into the prioritized operational space control framework.

### 3.3 Whole-body Impedance Controllers

In this chapter, we use our virtual quadruped-on-wheel robot (shown in Fig. 3.1) as an example to test our approach. For the robot, each leg has 4 joints, including three leg-revolute joints and one wheel joint. To enable the robot with manipulation behaviors, one manipulator can be installed on its base. The impedance controllers are established to enable the robot to track task references on rough terrain with compliance. For our robot, we control  $K$  operational tasks, including the quadruped-CoM motion with index 1, the wheeled motion with index 2 (and the manipulator-end motion with index 3 if there are manipulation missions, see the dashed boxes in Fig. 2.1).

To track the quadruped CoM motion, one impedance controller for the quadruped-CoM translational momentum and angular momentum is designed. Since the angular position of the quadruped CoM has

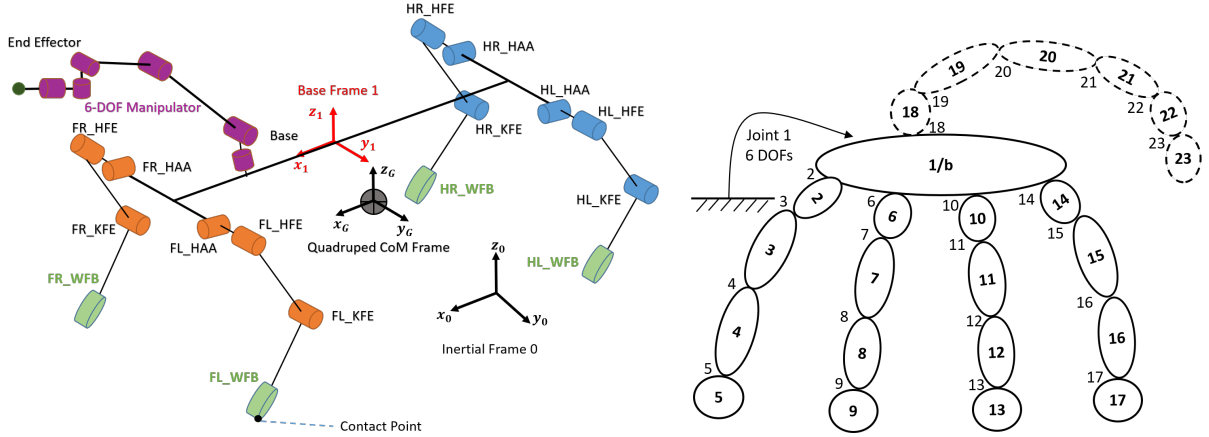


Figure 3.2: [Left:] Connectivity of TowerISIR: The robot has  $N_B = 23$  bodies, the manipulator with purple color accounts for 6 bodies and the quadruped part has 17 bodies. The quadruped-CoM frame  $G$  is set parallel with the base frame 1 or  $b$ . [Right:] Indexes from 1 to 17 represent the Quadruped Bodies. Dashed circles represent the manipulator with body/joint index from 18 to 23.

no sense, only the damping  $D_{G_\omega}$  for the angular momentum tracking is applied as follows,

$$\tau_{im_1} = \mathbf{J}_1^T \left[ \begin{array}{c} D_{G_\omega} (\mathbf{k}_G^d - \mathbf{k}_G) \\ \mathbf{K}_{G_v} (\mathbf{p}_G^d - \mathbf{p}_G) + D_{G_v} (\mathbf{v}_G^d - \mathbf{v}_G) \end{array} \right] \in \mathbb{R}^n, \quad (3.4)$$

where  $\mathbf{k}_G$  denotes the centroidal angular momentum.  $\mathbf{p}_G$  and  $\mathbf{v}_G$  represent the centroidal position and the centroidal translational velocity. In this section, the superscript  $d$  means “desired”.  $\mathbf{J}_1$  is the Jacobian for the centroidal motion.  $\mathbf{K}_{G_v}$  and  $D_{G_v}$  represent the stiffness and damping whose subscripts represent the CoM translational motion. For wheel locomotion, since the rolling resistance is not modeled, one impedance controller is built in the wheel joint space to compensate the inaccuracy of the contact model,

$$\tau_{im_2} = \mathbf{J}_2^T [\mathbf{K}_w (\mathbf{q}_w^d - \mathbf{q}_w) + \mathbf{D}_w (\dot{\mathbf{q}}_w^d - \dot{\mathbf{q}}_w)] \in \mathbb{R}^n, \quad (3.5)$$

where  $\mathbf{q}_w$  represents the wheel joint position.  $\mathbf{J}_2$  is the Jacobian for the wheel motion and it is also one selection matrix mapping vectors from  $\mathbb{R}^{n_w}$  to  $\mathbb{R}^n$  dimension.  $\mathbf{K}_w$  and  $\mathbf{D}_w$  are the stiffness and damping matrices respectively for the wheel motion. The manipulator enables the robot to achieve various missions. To achieve it with compliance, one impedance controller is added at the end effector of the manipulator.

$$\tau_{im_3} = \mathbf{J}_3^T [\mathbf{K}_m (\mathbf{q}_m^d - \mathbf{q}_m) + \mathbf{D}_m (\dot{\mathbf{q}}_m^d - \dot{\mathbf{q}}_m)] \in \mathbb{R}^n, \quad (3.6)$$

where  $\mathbf{q}_m$  can be achieved by inverse kinematics.  $\mathbf{J}_3$  is the Jacobian for the manipulator motion.  $\mathbf{K}_m$  and  $\mathbf{D}_m$  are the stiffness and damping matrices respectively. It is noticed that (3.4), (3.5) and (3.6) are impedance controllers for the designed three tasks, respectively, shown in Fig. 2.1. Then we add some extra impedance controllers which will indirectly influence the whole-body motion.

When running on the rough terrain, to enable the base with small angular motion, one impedance controller is designed,

$$\tau_{im_{bw}} = \mathbf{J}_{bw}^T [\mathbf{K}_{bw} (\boldsymbol{\theta}_b^d - \boldsymbol{\theta}_b) + \mathbf{D}_{bw} (\boldsymbol{\omega}_b^d - \boldsymbol{\omega}_b)] \in \mathbb{R}^n, \quad (3.7a)$$

$$\tau_{im_{bv}} = \mathbf{J}_{bv}^T [\mathbf{K}_{bv} (\mathbf{p}_b^d - \mathbf{p}_b) + \mathbf{D}_{bv} (\dot{\mathbf{p}}_b^d - \dot{\mathbf{p}}_b)] \in \mathbb{R}^n, \quad (3.7b)$$

where  $\mathbf{J}_{bw}$  and  $\mathbf{J}_{bv}$  denote the Jacobians for the base angular and translational motions,  $\boldsymbol{\omega}_b$  represents the base angular velocity.  $\mathbf{K}_{bw}$  and  $\mathbf{D}_{bw}$  are the stiffness and damping matrices for the base angular motion, and  $\mathbf{K}_{bv}$  and  $\mathbf{D}_{bv}$  are the ones for the base translational motion. In addition, to enable active legged suspension, another impedance controller for the legs is built as below,

$$\tau_{im_g} = \mathbf{J}_g^T [\mathbf{K}_g (\mathbf{p}_g^d - \mathbf{p}_g) + \mathbf{D}_g (\dot{\mathbf{p}}_g^d - \dot{\mathbf{p}}_g)] \in \mathbb{R}^n, \quad (3.8)$$

where  $\mathbf{p}_g \in \mathbb{R}^4$  combines all wheel-center positions relative to the base frame,  $\mathbf{J}_g$  is the relative Jacobian.  $\mathbf{K}_g$  and  $\mathbf{D}_g$  denote the stiffness and damping for legged suspension.

To be more general, more general impedance controllers can be built using the potential field as below,

$$\boldsymbol{\tau}_{im_p} = -\mathbf{S}_t^T \left[ \left( \frac{\partial \mathbf{V}_t(\mathbf{q})}{\partial \mathbf{q}} \right)^T + \mathbf{D}_t(\mathbf{q})\dot{\mathbf{q}} \right] \in \mathbb{R}^n, \quad (3.9)$$

where  $\mathbf{S}_t$  is one general selection matrix.  $\mathbf{V}_t(\mathbf{q})$  describes a repulsive potential field. The damping is injected by the positive, semi-definite matrix  $\mathbf{D}_t(\mathbf{q})$  and the subscript represents any compliance mission, especially for safety and physical constraints, such as self-collision and obstacle-avoidance missions [Sentis 2005] [Dietrich 2012]. For example, when an obstacle is approaching the head of one humanoid robot in [Sentis 2005], the reactive force generated by 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 chapter, we propose to enable this reactive force to influence whole-body motion by respecting task hierarchies. In this section, (3.7), (3.8) and (3.9) serve as the general impedance controllers which indirectly influence the designed specific tasks, c.f. Fig. 3.1.

### 3.4 Prioritized Impedance Controller based Torque Control

In this section, we will introduce how to add direct and indirect impedance controllers into four kinds of hierarchical torque controllers, and how they influence the designed tasks in a decoupled way. The torque controllers include one pseudo-inverse dynamics method, one null-space based weighted convex optimization solver, one hierarchical optimization solver and one hybrid hierarchical-weighted optimization solver.

#### 3.4.1 Modified Dynamics Model for Pseudo-inverse based Dynamics

We propose to integrate various impedance controllers into the hierarchical control framework. All impedance controllers should be consistent with task priorities and we define the prioritized total impedance control torque as below,

$$\boldsymbol{\Gamma} = \left[ \sum_{t=1}^K \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T \right] \sum \boldsymbol{\tau}_{im_j} \in \mathbb{R}^n, \quad (3.10)$$

where  $\sum \boldsymbol{\tau}_{im_j}$  sums all the impedance torques in Section 3.3. We call (3.10) the prioritized impedance controller,  $\boldsymbol{\Gamma}$  which relates to all  $K$  hierarchies. The prioritized impedance force  $\boldsymbol{\Gamma}$  respects the task priorities and the contact constraints. The operational space impedance force for hierarchy  $i$  can be derived as follows,

$$\bar{\mathbf{J}}_{i|pre(i)}^T \boldsymbol{\Gamma} = \bar{\mathbf{J}}_{i|pre(i)}^T \sum \boldsymbol{\tau}_{im_j}, \quad (3.11)$$

where we use the property  $\bar{\mathbf{J}}_{i|pre(i)}^T \bar{\mathbf{J}}_{j|pre(j)}^T = \mathbf{0}$  ( $i \neq j$ ). It means that the operational space impedance force for hierarchy  $i$  is decoupled from the other hierarchies. The generalized dynamics equation in (3.1) can be modified to integrate the prioritized sum impedance controller  $\boldsymbol{\Gamma}$  as follows,

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G} = \mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau} - \boldsymbol{\Gamma}, \quad (3.12)$$

which  $\boldsymbol{\Gamma}$  can be treated as the compensation item of the inaccurate dynamics model. It is noticed that this modified model is just used to develop our torque controllers in this subsection and in the subsection 3.4.2. By using (3.12), the calculated actuated torque can naturally combine the impedance controllers. For the pseudo-inverse based dynamics method [Sentis 2005], the actuated torque can be derived as below,

$$\boldsymbol{\tau} = \underbrace{(\mathbf{P}_{QR}\mathbf{S}^T)^+ \mathbf{P}_{QR}}_{\mathbf{W}} \left[ \boldsymbol{\Gamma} + \sum_{i=1}^K \bar{\mathbf{J}}_{i|pre(i)}^T \mathbf{F}_{i|pre(i)} \right], \quad (3.13)$$

where  $\mathbf{F}_{i|pre(i)}$  represents the operational space force for task  $i$  which is in the null space of tasks  $< i$ . The actuated torque  $\boldsymbol{\tau}$  is derived by the QR decomposition projector,  $\mathbf{P}_{QR}$  enables the dynamics model to be independent of the contact force and it is defined in [Mistry 2010]. In this section, the prioritized impedance controller is integrated into the pseudo-inverse based dynamics method. In the following sections, we will integrate the prioritized impedance controller into three optimization based torque controllers.

### 3.4.2 Prioritized Impedance Controller based Null-space based Prioritized Weighted Optimization (P-WQP)

In conventional weighted convex optimization [Bouyarmane 2011][Salini 2011], conventional feedback based controllers influence all sub-tasks together which cannot be accepted. In contrast with the state of the art, for our null-space weighted convex optimization developed in [Du 2020a], the hierarchies still exist among all the sub-tasks. Although this weighted scheme cannot ensure strict hierarchies, this null-space based weighted solver enable us to use the prioritized impedance controller to integrate various impedance controllers to influence the sub-tasks in a decoupled way. The cost function  $Q(\boldsymbol{\tau})$  in the operational space can be designed as follows,

$$Q(\boldsymbol{\tau}) = \|\ddot{\mathbf{x}}^* - \ddot{\mathbf{x}}\|_{\mathbf{w}}^2 + \varepsilon \|\boldsymbol{\tau}\|^2, \quad (3.14)$$

where  $\mathbf{w}$  is a weight matrix. The objective is regularized by a small value  $\varepsilon$ , which ensures positive definiteness of the objective Hessian. Using the modified dynamics model with prioritized impedance controllers in (3.12), the operational space force for the task  $i$  can be expressed as below,

$$\mathbf{F}_{i|pre(i)} = \boldsymbol{\Lambda}_{i|pre(i)} \ddot{\mathbf{x}}_i + \mathbf{h}_{i|pre(i)} - \mathbf{T}_{i|pre(i)} \quad (3.15a)$$

$$= \boldsymbol{\Lambda}_{i|pre(i)} \ddot{\mathbf{x}}_i + \mathbf{h}_{i|pre(i)} - \boldsymbol{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \mathbf{F}_{t|pre(t)} \quad (3.15b)$$

$$= \boldsymbol{\Lambda}_{i|pre(i)} \ddot{\mathbf{x}}_i + \mathbf{h}_{i|pre(i)} - \boldsymbol{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T (\mathbf{S}^T \boldsymbol{\tau} - \boldsymbol{\Gamma}) \quad (3.15c)$$

$$= \bar{\mathbf{J}}_{i|pre(i)}^T (\mathbf{S}^T \boldsymbol{\tau} - \boldsymbol{\Gamma}), \quad (3.15d)$$

where  $\mathbf{T}_{i|pre(i)}$  combines the tasks with priorities higher than  $i$ . (3.15a) and (3.15d) are derived by multiplying  $\bar{\mathbf{J}}_{i|pre(i)}^T$  at respectively the left and right sides of the modified dynamics model in (3.12). Then the quadratic form of the task  $i$  acceleration extraction in (3.17) can be achieved using (3.15) and the following formula,

$$\ddot{\mathbf{x}}_i = \boldsymbol{\Lambda}_{i|pre(i)}^{-1} [\mathbf{F}_{i|pre(i)} + \mathbf{T}_{i|pre(i)} - \mathbf{h}_{i|pre(i)}] = \mathbf{A}_i^T \boldsymbol{\tau} + \mathbf{B}_i. \quad (3.16)$$

where we define  $\mathbf{h}_{i|pre(i)}$  represents the sum of the Coriolis & centrifugal force and the gravity force in the task  $i$  operational space.  $\mathbf{h}_{i|pre(i)}$  is in the null spaces of tasks with priorities higher than  $i$ , detailed in [Park 2006].  $\mathbf{A}_i$  and  $\mathbf{B}_i$  can be derived using the modified dynamics model in (3.12) as follows,

$$\mathbf{A}_i^T = \mathbf{J}_{i|pre(i)} \mathbf{M}^{-1} \mathbf{S}^T + \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T \mathbf{S}^T, \quad (3.17a)$$

$$\mathbf{B}_i = -\mathbf{J}_{i|pre(i)} \mathbf{M}^{-1} \boldsymbol{\Gamma} - \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T \boldsymbol{\Gamma} - \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \mathbf{h}_{i|pre(i)}. \quad (3.17b)$$

In addition,  $\ddot{\mathbf{x}}$  combines all sub-task accelerations as below,

$$\ddot{\mathbf{x}} = \begin{bmatrix} \ddot{\mathbf{x}}_1 \\ \vdots \\ \ddot{\mathbf{x}}_K \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^T \\ \vdots \\ \mathbf{A}_K^T \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} = \mathbf{A}^T \boldsymbol{\tau} + \mathbf{B}. \quad (3.18)$$



The common inequality constraints for this framework (such as the actuation power constraints, the base orientation and position limits, the actuated-joint position and velocity limits, the unilateral contact constraint and the friction cone) are formulated as below,

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_{|k} \leq \boldsymbol{\tau}_{max} \quad (3.19a)$$

$$\mathbf{p}_{b|min} \leq \mathbf{p}_{b|k+1} = \mathbf{p}_{b|k} + \dot{\mathbf{p}}_{b|k}\delta t + \frac{1}{2}\ddot{\mathbf{p}}_{b|k}\delta t^2 \leq \mathbf{p}_{b|max} \quad (3.19b)$$

$$\boldsymbol{\theta}_{b|min} \leq \boldsymbol{\theta}_{b|k+1} = \boldsymbol{\theta}_{b|k} + \dot{\boldsymbol{\theta}}_{b|k}\delta t + \frac{1}{2}\ddot{\boldsymbol{\theta}}_{b|k}\delta t^2 \leq \boldsymbol{\theta}_{b|max} \quad (3.19c)$$

$$\mathbf{q}_{s|min} \leq \mathbf{q}_{s|k+1} = \mathbf{q}_{s|k} + \dot{\mathbf{q}}_{s|k}\delta t + \frac{1}{2}\ddot{\mathbf{q}}_{s|k}\delta t^2 \leq \mathbf{q}_{s|max} \quad (3.19d)$$

$$\dot{\mathbf{q}}_{s|min} \leq \dot{\mathbf{q}}_{s|k+1} = \dot{\mathbf{q}}_{s|k} + \ddot{\mathbf{q}}_{s|k}\delta t \leq \dot{\mathbf{q}}_{s|max} \quad (3.19e)$$

$$\mathbf{u}_c \cdot \mathbf{F}_c \geq \mathbf{0}, \quad (3.19f)$$

$$|\mathbf{t}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c) \quad (3.19g)$$

$$|\mathbf{b}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c), \quad (3.19h)$$

where  $|k$  and  $\delta t$  denote the  $k$ th instance and the fixed sampling time, respectively.  $\mu$  is the friction coefficient.  $(\mathbf{u}_c, \mathbf{t}_c, \mathbf{b}_c)$  are used to select the normal, tangential and lateral directions of the contact force.  $\dot{\mathbf{p}}_b, \ddot{\mathbf{p}}_b, \dot{\boldsymbol{\theta}}_b$  and  $\ddot{\boldsymbol{\theta}}_b$  are achieved in Appendix D. For our robot with one manipulator, there are total 92 inequality constraints, which includes  $n_a$  joint actuation limits in (3.19a), 6 constraints relating to the base position and orientation in (3.19b) and (3.19c),  $n_a$  actuated-joint position limits in (3.19d),  $n_a$  actuated-joint velocity limits in (3.19e), and 4 constraints for unilateral conditions in (3.19f), 16 constraints relating to the friction cone in (3.19g) and (3.19h).

The quadratic forms of the contact force  $\mathbf{F}_c$  and the joint acceleration  $\ddot{\mathbf{q}}$  are extracted using our dynamics model with prioritized impedance controllers in (3.12) as below,

$$\mathbf{F}_c = -\bar{\mathbf{J}}_c^T \mathbf{S}^T \boldsymbol{\tau} + \mathbf{h}_c + \bar{\mathbf{J}}_c^T \boldsymbol{\Gamma} = \mathbf{A}_c^T \boldsymbol{\tau} + \mathbf{B}_c, \quad (3.20a)$$

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1} (\mathbf{J}_c^T \mathbf{A}_c^T + \mathbf{S}^T) \boldsymbol{\tau} + \mathbf{M}^{-1} (\mathbf{J}_c^T \mathbf{B}_c - \mathbf{C} - \mathbf{G} - \boldsymbol{\Gamma}) = \mathbf{A}_q^T \boldsymbol{\tau} + \mathbf{B}_q, \quad (3.20b)$$

where  $\bar{\mathbf{J}}_c$  is the dynamically generalized inverse of contact Jacobian.  $\mathbf{h}_c$  denotes the sum of the Coriolis & centrifugal and the gravity forces at the contact points. It is noticed that  $\bar{\mathbf{J}}_c^T \boldsymbol{\Gamma} = \mathbf{0}$ . In this way, the inequality constraints can be expressed depending on the actuated torque and prioritized impedance forces by substituting the quadratic extractions in (3.20) into (3.19b)-(3.19h), noting that  $\ddot{\mathbf{q}}_s = \mathbf{S}\ddot{\mathbf{q}}$ . In addition, the control objective and constraints in (3.14) and (3.19) may not have a common solution, a trade-off can be expressed in form of slacks on the inequality constraints in (3.19), detailed in [Herzog 2016].

Compared with conventional weighted optimization, our method presents several advanced advantages: First, when singularity exists for one specific sub-task  $j$  in this hierarchy, this null-space based weighted scheme enables us to apply the eigen-decomposition method in [Sentis 2005] to select the feasible motion directions. In addition, our prioritized impedance controllers influence the sub-tasks in a decoupled way. We define P-WQP to represent this prioritized impedance controller based weighted QP solver. The null-space based weighted scheme can be integrated into the hierarchical control framework, detailed in Section 3.4.4.

### 3.4.3 One New Dynamics Model for Prioritized Impedance Controller based Hierarchical Optimization (P-HQP)

We proposed in [Du 2020f] one general hierarchical optimization solver and one recursively updated dynamics model for each hierarchy optimization. In this chapter, we will extend and improve the solver in several aspects. We propose one new dynamics model for all hierarchies, which is more apparent and general. Then we integrate the prioritized impedance controller into this model, which enables to

simplify the task acceleration extraction for each hierarchy, especially for one hierarchy with several sub-tasks. The quadratic forms of the contact force and the joint acceleration are extracted depending on our prioritized impedance controller, which enables the inequality constraints to be compatible with each-hierarchy optimization solver. In this section, we extract the complete quadratic-form acceleration for one hierarchy with only one task. In next section, we will use the model for one hierarchy with more than one task and give the complete sub-task acceleration extraction. Finally, we develop one complete algorithm combining our prioritized impedance controller. Initially, we define P-HQP to represent the prioritized impedance controller based hierarchical QP solver in this subsection, and its control framework is shown in Fig. 3.3, in which we focus on P-HQP controller in the dashed red box.

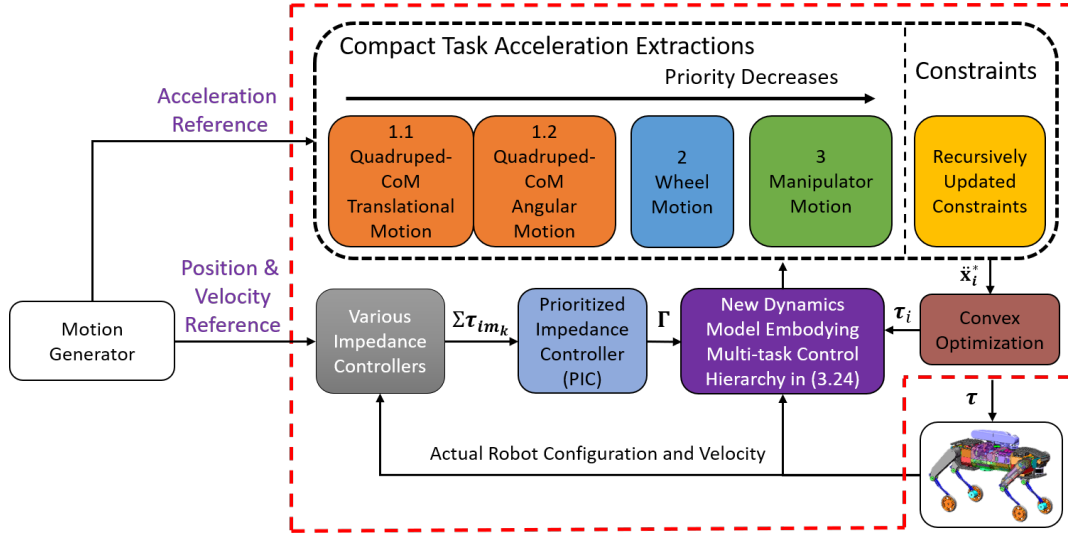


Figure 3.3: P-HQP control framework: Given the quadruped-CoM translational motion reference, the legged and wheeled motions are achieved, detailed in Chapter 4. The motion references serve as the inputs of our dynamics controller in the dashed red box, including the proposed modified dynamics model, task extractions, prioritized impedance controllers, recursively updated inequality constraints and convex optimization. Three hierarchies are defined and each color stands for one hierarchy. The variables in this framework are explained in Chapter 2 and in this chapter.

In the conventional method in [Escande 2014]-[Dietrich 2012], the direct impedance controller applying on the task can be added to achieve the task reference as follow,

$$\ddot{\mathbf{x}}_i^* = \ddot{\mathbf{x}}_i^d + [\mathbf{K}_i (\mathbf{x}_i^d - \mathbf{x}_i) + \mathbf{D}_i (\dot{\mathbf{x}}_i^d - \dot{\mathbf{x}}_i)], \quad (3.21)$$

where  $\mathbf{x}_i^d$  represents the desired task motion.  $\mathbf{K}_i$  and  $\mathbf{D}_i$  denote the relative stiffness and damping. However, when the designed tasks account all the robot DOF, the methods in the state of the art using (3.21) do not deal with the general impedance controllers which indirectly influence the task. Therefore, for this issue, in this section, we explain how to integrate general impedance controllers into the hierarchical convex optimization using our new dynamics model.

For  $K$  hierarchies, there are  $K$  desired acceleration references in the operational space, from  $\tilde{\mathbf{x}}_1^*$  to  $\tilde{\mathbf{x}}_K^*$ . The error between the task references  $\tilde{\mathbf{x}}_i^*$  and expected  $\tilde{\mathbf{x}}_i$  can be used as inputs of the optimization solver. The hierarchical convex optimization in operational space can be designed for each hierarchy simply as,

$$Q_i(\boldsymbol{\tau}_i) = \|\tilde{\mathbf{x}}_i^* - \tilde{\mathbf{x}}_i\|^2 + \varepsilon \|\boldsymbol{\tau}_i\|^2, \quad (3.22)$$

where  $Q_i(\boldsymbol{\tau}_i)$  is the cost function.  $\boldsymbol{\tau}_i \in \mathbb{R}^{n_a}$  is the optimized required torque for task  $i$  which will not conflict the achievement of tasks with priorities higher than level  $i$ .



According to the prioritized operational space control framework, the following torque equation embodies a multi-level control hierarchy:

$$\mathbf{S}^T \boldsymbol{\tau} = \sum_{i=1}^K \mathbf{N}_{pre(i)}^T \mathbf{S}^T \boldsymbol{\tau}_i, \quad (3.23)$$

where  $K$  is the hierarchy number. We propose to modify the generalized dynamics model in (3.12) by substituting (3.23) into it as follows,

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C} + \mathbf{G} = \mathbf{J}_c^T \mathbf{F}_c - \boldsymbol{\Gamma} + \sum_{i=1}^K \mathbf{N}_{pre(i)}^T \mathbf{S}^T \boldsymbol{\tau}_i, \quad (3.24)$$

It is noticed that this modified model is also used to develop the hierarchical torque controller in this subsection and the more complex hybrid control scheme in the subsection 3.4.4. This model can be used any hierarchy. Using the recursively updated dynamics model with prioritized impedance controllers in (3.24), the operational space force for task  $i$  can be expressed as below,

$$\mathbf{F}_{i|pre(i)} = \boldsymbol{\Lambda}_{i|pre(i)} \ddot{\mathbf{x}}_i + \mathbf{h}_{i|pre(i)} - \mathbf{T}_{i|pre(i)} \quad (3.25a)$$

$$= \boldsymbol{\Lambda}_{i|pre(i)} \ddot{\mathbf{x}}_i + \mathbf{h}_{i|pre(i)} - \boldsymbol{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \mathbf{F}_{t|pre(t)} \quad (3.25b)$$

$$= \bar{\mathbf{J}}_{i|pre(i)}^T \left( -\boldsymbol{\Gamma} + \sum_{t=1}^K \mathbf{N}_{pre(t)}^T \mathbf{S}^T \boldsymbol{\tau}_t \right) \quad (3.25c)$$

$$= \bar{\mathbf{J}}_{i|pre(i)}^T \left( \mathbf{S}^T \boldsymbol{\tau}_i - \boldsymbol{\Gamma} + \sum_{t=1}^{i-1} \mathbf{S}^T \boldsymbol{\tau}_t \right), \quad (3.25d)$$

where (3.25a) and (3.25c) are derived by multiplying  $\bar{\mathbf{J}}_{i|pre(i)}^T$  at respectively the left and right sides of the dynamics model in (3.24). Then the quadratic form of the task  $i$  acceleration extraction in (3.27) can be achieved by combining (3.16) and (3.25) depending on the prioritized impedance control torque  $\boldsymbol{\Gamma}$  as below,

$$\ddot{\mathbf{x}}_i = \mathbf{A}_i^T \boldsymbol{\tau}_i + \mathbf{B}_i, \quad (3.26)$$

where the general matrices  $\mathbf{A}_i$  and  $\mathbf{B}_i$  are derived as follows,

$$\mathbf{A}_i^T = \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \bar{\mathbf{J}}_{i|pre(i)}^T \mathbf{S}^T, \quad (3.27a)$$

$$\mathbf{B}_i = \mathbf{J}_{i|pre(i)} \mathbf{M}^{-1} \left[ -\boldsymbol{\Gamma} + \sum_{t=1}^{i-1} \mathbf{S}^T \boldsymbol{\tau}_t \right] - \boldsymbol{\Lambda}_{i|pre(i)}^{-1} (\mathbf{h}_{i|pre(i)} + \mathbf{T}_{i|pre(i)}), \quad (3.27b)$$

where  $\boldsymbol{\Lambda}_{i|pre(i)}$  represents the task  $i$  operational space inertia and  $\mathbf{T}_{i|pre(i)}$  combines all operational space forces of the task levels higher than  $i$ , detailed in [Sentis 2005],

$$\mathbf{T}_{i|pre(i)} = \boldsymbol{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \mathbf{F}_{t|pre(t)}, \quad (3.28)$$

where  $\mathbf{F}_{t|pre(t)}$  can be achieved as below,

$$\mathbf{F}_{t|pre(t)} = \bar{\mathbf{J}}_{t|pre(t)}^T \left( -\boldsymbol{\Gamma} + \sum_{j=1}^K \mathbf{N}_{pre(j)}^T \mathbf{S}^T \boldsymbol{\tau}_j \right) = \bar{\mathbf{J}}_{t|pre(t)}^T \left( -\boldsymbol{\Gamma} + \sum_{j=1}^t \mathbf{S}^T \boldsymbol{\tau}_j \right), \quad (3.29)$$

where we use the properties as follows,

$$\bar{\mathbf{J}}_{t|pre(t)}^T \mathbf{N}_{pre(j)} = \bar{\mathbf{J}}_{t|pre(t)}^T, \quad \text{if } j \leq t, \quad (3.30a)$$

$$= \mathbf{0}, \quad \text{if } j > t, \quad (3.30b)$$

In Appendix B, we explain the derivations of  $\mathbf{A}_i$  and  $\mathbf{B}_i$  for the hierarchical convex optimization. In this way, the prioritized impedance controller is integrated into the hierarchical convex optimization. Then (3.26) is derived and is substituted to (3.22) to achieve the optimized torque for the current hierarchy. It is noticed that in the beginning, the required torque for each task  $\tau_i$ ,  $i \in \{1, \dots, K\}$ , is set with zero vectors. Then in the hierarchy  $i$  optimization,  $(\tau_1, \dots, \tau_{i-1})$  are known,  $\tau_i$  is the optimized variable, and  $(\tau_{i+1}, \dots, \tau_K)$  are still zero vectors. Therefore, the actuated torque  $\tau$  in (3.23) can be expressed for hierarchy  $i$  as below,

$$\tau = \mathbf{W} \left[ \sum_{t=1}^i \mathbf{N}_{pre(t)}^T \mathbf{S}^T \tau_t \right] = \mathbf{W} \left[ \mathbf{N}_{pre(i)}^T \mathbf{S}^T \tau_i + \sum_{t=1}^{i-1} \mathbf{N}_{pre(t)}^T \mathbf{S}^T \tau_t \right], \quad (3.31)$$

where this actuated torque  $\tau$  combines the outputs of hierarchy  $\leq i$  and it is used for the inequality constraints in (3.19). In this additional recursive way, for the final hierarchy, the index  $i$  in (3.31) is equal to  $K$ .

For the hierarchy  $i$ , the quadratic form of the contact force can be rewritten according to the optimized variable  $\tau_i$  by substituting (3.31) into (3.20a) as follows,

$$\mathbf{F}_c = \mathbf{A}_{c_i}^T \tau_i + \mathbf{B}_{c_i}, \quad (3.32)$$

where  $\mathbf{A}_{c_i}$  and  $\mathbf{B}_{c_i}$  are derived combining (3.20a) and (3.31) as below,

$$\mathbf{A}_{c_i}^T = -\bar{\mathbf{J}}_c^T \mathbf{S}^T \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T, \quad (3.33a)$$

$$\mathbf{B}_{c_i} = \mathbf{h}_c + \bar{\mathbf{J}}_c^T \mathbf{\Gamma} - \bar{\mathbf{J}}_c^T \mathbf{S}^T \mathbf{W} \sum_{t=1}^{i-1} \mathbf{N}_{pre(t)}^T \mathbf{S}^T \tau_t, \quad (3.33b)$$

and the quadratic forms of the joint acceleration in (3.20b) can be rewritten as follows,

$$\ddot{\mathbf{q}} = \mathbf{A}_{q_i}^T \tau_i + \mathbf{B}_{q_i}, \quad (3.34)$$

where  $\mathbf{A}_{q_i}$  and  $\mathbf{B}_{q_i}$  are derived combining (3.20b) and (3.31) as below,

$$\mathbf{A}_{q_i}^T = \mathbf{M}^{-1} \left( \mathbf{J}_c^T \mathbf{A}_{c_i}^T + \mathbf{S}^T \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T \right), \quad (3.35a)$$

$$\mathbf{B}_{q_i} = \mathbf{M}^{-1} \left( \mathbf{J}_c^T \mathbf{A}_{c_i}^T + \mathbf{S}^T \right) \mathbf{W} \sum_{t=1}^{i-1} \mathbf{N}_{pre(t)}^T \mathbf{S}^T \tau_t + \mathbf{M}^{-1} \left( \mathbf{J}_c^T \mathbf{B}_{c_i} - \mathbf{C} - \mathbf{G} - \mathbf{\Gamma} \right), \quad (3.35b)$$

where  $(\mathbf{A}_{c_i}, \mathbf{B}_{c_i})$  and  $(\mathbf{A}_{q_i}, \mathbf{B}_{q_i})$  are used for the optimization of the hierarchy  $i$ . It is noticed that after all task hierarchies are solved, the final  $\mathbf{F}_c$  is the final contact force for all tasks. The base angular velocity in the float-base frame can be expressed as below,

$$\omega_b = \underbrace{\begin{bmatrix} 1 & 0 & -\sin(\beta) \\ 0 & \cos(\alpha) & \cos(\beta) \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \cos(\beta) \end{bmatrix}}_{\Theta} \dot{\theta}_b, \quad (3.36a)$$

$$\dot{\omega}_b = \dot{\Theta} \dot{\theta}_b + \Theta \ddot{\theta}_b, \quad (3.36b)$$

then  $\dot{\theta}_b$  and  $\ddot{\theta}_b$  can be achieved using the following relationships,

$$\dot{\theta}_b = \Theta^{-1} \omega_b, \quad (3.37a)$$

$$\ddot{\theta}_b = \Theta^{-1} \dot{\omega}_b - \Theta^{-1} \dot{\Theta} \dot{\theta}_b, \quad (3.37b)$$

The base translational velocity is derived as follows,

$$\dot{\mathbf{p}}_b = {}^0 \mathbf{R}_b \mathbf{v}_b, \quad (3.38a)$$

$$\ddot{\mathbf{p}}_b = {}^0 \mathbf{R}_b (\dot{\mathbf{v}}_b + \omega_b \times \mathbf{v}_b), \quad (3.38b)$$

where  ${}^0\mathbf{R}_b$  is the rotation matrix from the floating-base frame to the inertial frame.  $\mathbf{v}_b$  is the base translational velocity expressed in the floating-base frame.

In this way, the extractions in (3.32) and (3.34) can be substituted into (3.19b)-(4.18b) to construct the contact and joint constraints for the hierarchy  $i$  optimization framework. For this actuated-torque constraint, (3.31) can be substituted into (3.19a). Then the inequality constraints in (3.19) are rewritten as follows,

$$\boldsymbol{\tau}_{min} \leq \left[ \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T \right] \boldsymbol{\tau}_i + \left[ \mathbf{W} \sum_{t=1}^{i-1} \mathbf{N}_{pre(t)}^T \mathbf{S}^T \boldsymbol{\tau}_t \right] \leq \boldsymbol{\tau}_{max} \quad (3.39a)$$

$$\mathbf{p}_{b|min} \leq \left[ \frac{1}{2} \delta t^2 \mathbf{R}_b^T \mathbf{S}_b \mathbf{A}_{q_i}^T \right] \boldsymbol{\tau}_i + [\mathbf{p}_{b|k} + \dot{\mathbf{p}}_{b|k} \delta t + \boldsymbol{\Xi}_{b|k}] \leq \mathbf{p}_{b|max} \quad (3.39b)$$

$$\boldsymbol{\theta}_{b|min} \leq \left[ \frac{1}{2} \delta t^2 \boldsymbol{\Theta}^{-1} \mathbf{S}_\theta \mathbf{A}_{q_i}^T \right] \boldsymbol{\tau}_i + [\boldsymbol{\theta}_{b|k} + \dot{\boldsymbol{\theta}}_{b|k} \delta t + \boldsymbol{\Xi}_{\theta|k}] \leq \boldsymbol{\theta}_{b|max} \quad (3.39c)$$

$$\mathbf{q}_{s|min} \leq \left[ \frac{1}{2} \delta t^2 \mathbf{S} \mathbf{A}_{q_i}^T \right] \boldsymbol{\tau}_i + \left[ \frac{1}{2} \delta t^2 \mathbf{S} \mathbf{B}_{q_i} + \mathbf{q}_{s|k} + \dot{\mathbf{q}}_{s|k} \delta t \right] \leq \mathbf{q}_{s|max} \quad (3.39d)$$

$$\dot{\mathbf{q}}_{s|min} \leq [\delta t \mathbf{S} \mathbf{A}_{q_i}^T] \boldsymbol{\tau}_i + [\delta t \mathbf{S} \mathbf{B}_{q_i} + \dot{\mathbf{q}}_{s|k}] \leq \dot{\mathbf{q}}_{s|max} \quad (3.39e)$$

$$\mathbf{u}_c^T \mathbf{A}_{c_i}^T \boldsymbol{\tau}_i + \mathbf{u}_c^T \mathbf{B}_{c_i} \geq 0, \quad (3.39f)$$

$$\left[ -\frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{A}_{c_i}^T - \mathbf{t}_c^T \mathbf{A}_{c_i}^T \right] \boldsymbol{\tau}_i \leq \mathbf{t}_c^T \mathbf{B}_{c_i} + \frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{B}_{c_i} \quad (3.39g)$$

$$\left[ -\frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{A}_{c_i}^T + \mathbf{t}_c^T \mathbf{A}_{c_i}^T \right] \boldsymbol{\tau}_i \leq \frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{B}_{c_i} - \mathbf{t}_c^T \mathbf{B}_{c_i} \quad (3.39h)$$

$$\left[ -\frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{A}_{c_i}^T - \mathbf{b}_c^T \mathbf{A}_{c_i}^T \right] \boldsymbol{\tau}_i \leq \mathbf{b}_c^T \mathbf{B}_{c_i} + \frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{B}_{c_i} \quad (3.39i)$$

$$\left[ -\frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{A}_{c_i}^T + \mathbf{b}_c^T \mathbf{A}_{c_i}^T \right] \boldsymbol{\tau}_i \leq \frac{\mu}{\sqrt{2}} \mathbf{u}_c^T \mathbf{B}_{c_i} - \mathbf{b}_c^T \mathbf{B}_{c_i}, \quad (3.39j)$$

where  $\mathbf{S}_\theta$  and  $\mathbf{S}_b$  are used to select the first and second 3 rows of  $\ddot{\mathbf{q}}$ . (3.39g) and (3.39h) both constitute the inequality constraint in (3.19g). In addition, (3.39i) and (3.39j) are built to represent the constraint in (3.19h). In addition,  $\boldsymbol{\Xi}_b$  and  $\boldsymbol{\Xi}_\theta$  in (3.39b) and (3.39c) respectively are achieved as below,

$$\boldsymbol{\Xi}_b = \frac{1}{2} {}^0\mathbf{R}_b (\mathbf{S}_b \mathbf{B}_{q_i} + \boldsymbol{\omega}_b \times \mathbf{v}_b) \delta t^2, \quad (3.40a)$$

$$\boldsymbol{\Xi}_\theta = \frac{1}{2} \boldsymbol{\Theta}^{-1} (\mathbf{S}_\theta \mathbf{B}_{q_i} - \dot{\boldsymbol{\Theta}} \dot{\boldsymbol{\theta}}_b) \delta t^2. \quad (3.40b)$$

Compared with the recent hierarchical control solvers in [Bellicoso 2016], [Herzog 2016], [Kanoun 2011], [De Lasa 2010], this P-HQP solver combines the benefits of the four algorithms and we extract each task reference by taking full advantages of the null-space based multi-task operational space dynamics control scheme. This enables the several tasks to be decoupled, which is expressed in the control objective and the inequality constraints. In addition, the equality constraints are eliminated from one QP to the other, resulting in faster solvable QPs. For our approach, the inequality constraint number is the same for all hierarchies. However, the inequality constraints in [De Lasa 2010] are included only in the first priority. Instead, the methods in [Bellicoso 2016][Herzog 2016][Kanoun 2011] allow for prioritization among inequality constraints, then the constraints for the lower hierarchies are added into the higher hierarchies, and this enlarges the constraint number. Furthermore, our method is more general which is achieved by using the principle of one new dynamics model. Our method is sufficiently fast shown Section 3.5, which is also suitable for high-DOF robots, e.g. humanoids. With an active set method, it can outperform QP cascades in terms of speed. The open-source library in [Ferreau 2014] enables the use of warm-start techniques to speed up the computations. To be emphasized, as the focus of this chapter is the prioritized impedance controller, our approach enables to integrate it into the QP problem formulation easily, and it is implemented successfully in Section 3.5. Another practical advan-

tage of our method is the easily implemented regularization in face of task singularities by applying the eigen-decomposition method, as discussed in [Sentis 2005].

### 3.4.4 Integration of the Prioritized Weighted Scheme in the Hierarchical Optimization (P-HWQP)

In this section, the integration of prioritized impedance controller is researched for a more complex scenario. When tasks in one hierarchy need importance distribution and no priority can be decided, the weighted scheme in Section 3.4.2 should be applied. Assume that there are  $r_i$  sub-tasks in the hierarchy  $i$  and they should all be in the null space of tasks with higher priorities. The Jacobian for the  $r_i$  sub-tasks is written as,

$$\mathbf{J}_i = \left[ \mathbf{J}_{i_1}^T \quad \cdots \quad \mathbf{J}_{i_{r_i}}^T \right]^T, \quad (3.41)$$

where  $\mathbf{J}_{i_j}$  is the Jacobian for task  $j$  in hierarchy  $i$ . After the previous  $(i-1)$  hierarchies are calculated, the  $r_i$  tasks in hierarchy  $i$  will be optimized together and contribute to the current-hierarchy required torque. The required dynamics model for this weighted scheme becomes from (3.12) to (3.24). The prioritized impedance controller should be modified as follows,

$$\mathbf{\Gamma} = \left[ \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T + \sum_{p=1}^{r_i} \mathbf{J}_{i_p|pre(i_p)}^T \bar{\mathbf{J}}_{i_p|pre(i_p)}^T + \sum_{t=i+1}^K \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T \right] \sum \boldsymbol{\tau}_{im_k} \in \mathbb{R}^n, \quad (3.42)$$

where  $\mathbf{\Gamma}$  integrates the prioritized total impedance control torque for the strict hierarchies with indexes  $\leq (i-1)$  and  $\geq (i+1)$ , and the hierarchy  $i$  with  $r_i$  sub-tasks. Using the recursively updated dynamics model with prioritized impedance controllers in (3.24), the operational space force for task  $i$  can be expressed as below,

$$\mathbf{F}_{i_j|pre(i_j)} = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \ddot{\mathbf{x}}_{i_j} + \mathbf{h}_{i_j|pre(i_j)} - \mathbf{T}_{i_j|pre(i_j)} \quad (3.43a)$$

$$= \bar{\mathbf{J}}_{i_j|pre(i_j)}^T \left( \mathbf{S}^T \boldsymbol{\tau}_i - \mathbf{\Gamma} + \sum_{t=1}^{i-1} \mathbf{S}^T \boldsymbol{\tau}_t \right), \quad (3.43b)$$

where (3.43a) and (3.43b) are derived by multiplying  $\bar{\mathbf{J}}_{i_j|pre(i_j)}^T$  at respectively the left and right sides of the dynamics model in (3.24).  $\mathbf{T}_{i_j|pre(i_j)}$  combines the tasks with priorities higher than  $i$ , and the subtasks from  $i_1$  to  $i_{j-1}$ .

$$\begin{aligned} \mathbf{T}_{i_j|pre(i_j)} &= \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \left[ \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \mathbf{F}_{t|pre(t)} + \sum_{p=1}^{j-1} \mathbf{J}_{i_p|pre(i_p)}^T \mathbf{F}_{i_p|pre(i_p)} \right] \\ &= \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \left[ \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T (-\mathbf{\Gamma} + \sum_{k=1}^t \mathbf{S}^T \boldsymbol{\tau}_k) \right] + \\ &\quad \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \left[ \sum_{p=1}^{j-1} \mathbf{J}_{i_p|pre(i_p)}^T \bar{\mathbf{J}}_{i_p|pre(i_p)}^T (\mathbf{S}^T \boldsymbol{\tau}_i - \mathbf{\Gamma} + \sum_{t=1}^{i-1} \mathbf{S}^T \boldsymbol{\tau}_t) \right] \\ &= \mathbf{U}_{i_j} \boldsymbol{\tau}_i + \mathbf{V}_{i_j}, \end{aligned} \quad (3.44)$$

where  $\mathbf{U}_{i_j}$  and  $\mathbf{V}_{i_j}$  can be extracted as follows,

$$\mathbf{U}_{i_j} = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \sum_{p=1}^{j-1} \mathbf{J}_{i_p|pre(i_p)}^T \bar{\mathbf{J}}_{i_p|pre(i_p)}^T \mathbf{S}^T, \quad (3.45a)$$

$$\begin{aligned} \mathbf{V}_{i_j} &= \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \left[ \sum_{t=1}^{i-1} \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T (-\mathbf{\Gamma} + \sum_{k=1}^t \mathbf{S}^T \boldsymbol{\tau}_k) \right] + \\ &\quad \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \left[ \sum_{p=1}^{j-1} \mathbf{J}_{i_p|pre(i_p)}^T \bar{\mathbf{J}}_{i_p|pre(i_p)}^T (-\mathbf{\Gamma} + \sum_{t=1}^{i-1} \mathbf{S}^T \boldsymbol{\tau}_t) \right]. \end{aligned} \quad (3.45b)$$



**Algorithm 1** The Hybrid Hierarchical-Weighted Convex Optimization (P-HWQP)

P-HWQP in Section 3.4.4 combines P-WQP in Section 3.4.2 and P-HQP in Section 3.4.3 together, and the developed algorithm is presented by introducing several middle-process variables.

**Inputs:**

User-defined inputs:  $K$  hierarchies. Each hierarchy has  $r_i \geq 1$  sub-tasks.

Desired motion inputs: All task acceleration references from motion generators.

Instantaneous feedback inputs: Joint position  $\mathbf{q}$  and velocity  $\dot{\mathbf{q}}$  from sensors.

**Re-evaluated parameters:**

Generalized dynamics parameters:  $\mathbf{M}$ ,  $\mathbf{C}$ ,  $\mathbf{G}$ .

Prioritized Impedance Controller:  $\mathbf{\Gamma}$ .

**Middle-process variables:**  $\mathbf{L}_i$  and  $\mathbf{H}_i$ ,  $i \in \{0, \dots, K\}$  to denote the sum of the torque in the actuated joint space and in the generalized coordinate space.  $\mathbf{L}_0 = \mathbf{0}$  and  $\mathbf{H}_0 = \mathbf{0}$ .

**Initialization:**  $\boldsymbol{\tau} = \mathbf{0}$ ,  $\boldsymbol{\tau}_i = \mathbf{0}$ ,  $i \in \{1, \dots, K\}$ .

**for**  $i=1$  **to**  $K$  **do**

Re-evaluate the extractions of  $\mathbf{F}_c$  and  $\ddot{\mathbf{q}}$ :  $\mathbf{A}_{c_i}$ ,  $\mathbf{B}_{c_i}$ ,  $\mathbf{A}_{q_i}$ ,  $\mathbf{B}_{q_i}$ .

**if** hierarchy  $i$  has 1 task **then**

% hierarchy  $i$  has 1 task. Apply null-space based hierarchical control scheme.

Re-evaluate the task parameters:  $\mathbf{J}_i$ ,  $\mathbf{J}_{i|pre(i)}$ ,  $\bar{\mathbf{J}}_{i|pre(i)}$ ,  $\boldsymbol{\Lambda}_{i|pre(i)}$ ,  $\mathbf{h}_{i|pre(i)}$ .

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

$$\mathbf{B}_i = \mathbf{J}_{i|pre(i)} \mathbf{M}^{-1} (\mathbf{S}^T \mathbf{L}_{i-1} - \mathbf{\Gamma}) - \boldsymbol{\Lambda}_{i|pre(i)}^{-1} \mathbf{h}_{i|pre(i)} + \mathbf{J}_i \mathbf{M}^{-1} \mathbf{H}_{i-1}$$

$$\boldsymbol{\tau}_i = \arg \min_{\boldsymbol{\tau}_i} \|\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i\|^2 + \varepsilon \|\boldsymbol{\tau}_i\|^2$$

s.t. the inequality constraints in (3.39)

$$\mathbf{L}_i = \mathbf{L}_{i-1} + \boldsymbol{\tau}_i$$

$$\mathbf{F}_{i|pre(i)} = \bar{\mathbf{J}}_{i|pre(i)}^T (\mathbf{S}^T \mathbf{L}_i - \mathbf{\Gamma})$$

$$\mathbf{H}_i = \mathbf{H}_{i-1} + \mathbf{J}_{i|pre(i)}^T \mathbf{F}_{i|pre(i)}$$

$$\boldsymbol{\tau} = \boldsymbol{\tau} + \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T \boldsymbol{\tau}_i$$

**else**

% hierarchy  $i$  has  $r_i$  sub-tasks. Apply the null-space based weighted control scheme.

**Initialization of middle-process variables:**  $\mathbf{P}_j$ ,  $j \in \{0, \dots, r_i\}$ ,  $\mathbf{P}_0 = \mathbf{0}$ .

Re-evaluate the sub-task parameters:  $\mathbf{J}_{i_j}$ ,  $\mathbf{J}_{i_j|pre(i_j)}$ ,  $\bar{\mathbf{J}}_{i_j|pre(i_j)}$ ,  $\boldsymbol{\Lambda}_{i_j|pre(i_j)}$ ,  $\mathbf{h}_{i_j|pre(i_j)}$ .

**for**  $j = 1$  **to**  $r_i$  **sub-task do**

$$\mathbf{U}_{i_j} = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \mathbf{P}_{j-1} \mathbf{S}^T$$

$$\mathbf{V}_{i_j} = \boldsymbol{\Lambda}_{i_j|pre(i_j)} \mathbf{J}_{i_j} \mathbf{M}^{-1} \{\mathbf{H}_{i-1} + \mathbf{P}_{j-1} (\mathbf{S}^T \mathbf{L}_{i-1} - \mathbf{\Gamma})\}$$

$$\mathbf{A}_{i_j}^T = \mathbf{J}_{i_j|pre(i_j)} \mathbf{M}^{-1} \mathbf{S}^T + \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1} \mathbf{U}_{i_j}$$

$$\mathbf{B}_{i_j} = \mathbf{J}_{i_j|pre(i_j)} \mathbf{M}^{-1} (\mathbf{S}^T \mathbf{L}_{i-1} - \mathbf{\Gamma}) + \boldsymbol{\Lambda}_{i_j|pre(i_j)}^{-1} \{\mathbf{V}_{i_j} - \mathbf{h}_{i_j|pre(i_j)}\}$$

$$\mathbf{P}_j = \mathbf{P}_{j-1} + \mathbf{J}_{i_j|pre(i_j)}^T \bar{\mathbf{J}}_{i_j|pre(i_j)}^T$$

**end**

$$\mathbf{A}_i = [\mathbf{A}_{i_1}, \dots, \mathbf{A}_{i_{r_i}}] \text{ and } \mathbf{B}_i = [\mathbf{B}_{i_1}^T, \dots, \mathbf{B}_{i_{r_i}}^T]^T$$

$$\boldsymbol{\tau}_i = \arg \min_{\boldsymbol{\tau}_i} \|\ddot{\mathbf{x}}_i^* - \ddot{\mathbf{x}}_i\|_{\mathbf{w}_i}^2 + \varepsilon \|\boldsymbol{\tau}_i\|^2$$

s.t. the inequality constraints in (3.39)

$$\mathbf{L}_i = \mathbf{L}_{i-1} + \boldsymbol{\tau}_i$$

$$\mathbf{H}_i = \mathbf{H}_{i-1} + \mathbf{P}_{r_i} (\mathbf{S}^T \mathbf{L}_i - \mathbf{\Gamma})$$

$$\boldsymbol{\tau} = \boldsymbol{\tau} + \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T \boldsymbol{\tau}_i$$

**end**

**end**

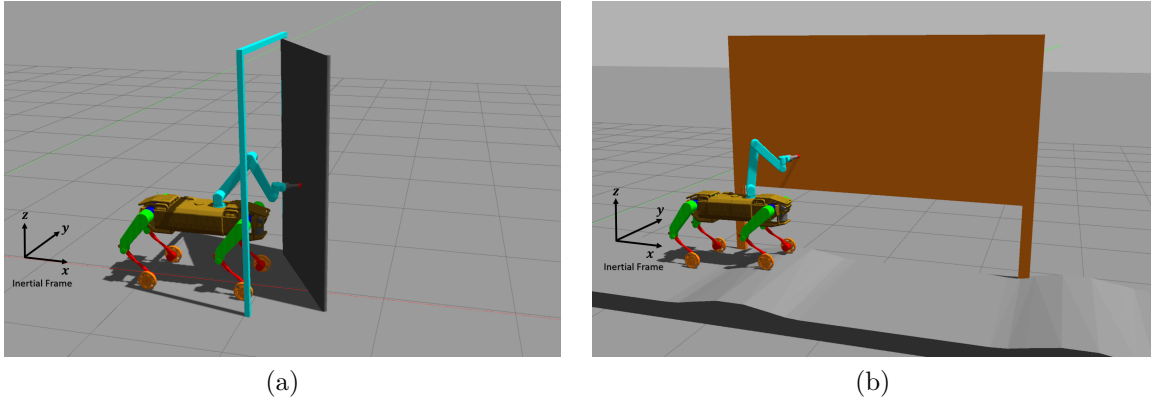


Figure 3.4: Two simulation scenarios under GAZEBO simulator. (a) The robot runs along with pushing a door open, including the extension and retraction of the manipulator. (b) The robot runs on the rough terrain along with washing a wall, including the suspension of the legs and the manipulator motion.

momentum model. The wheel acceleration can be derived using the contact kinematic acceleration constraint and the quadruped centroidal dynamics constraint. The motion generators are detailed in Chapter 4.

In subsection 3.5.1, the P-WQP control scheme is verified using *TowrISIR* with the manipulator on its floating base. In subsection 3.5.2, we use one more complex scenario to test our P-HWQP controller. The pseudo-inverse dynamics scheme with our prioritized impedance controller is applied in the first two simulations in Chapter 4, and the P-HQP controller is verified in the second two simulations in Chapter 4 and simulations in Chapter 5. Since the null-space based theory can handle the priorities among several hierarchies in the feed-forward torque controller and the prioritized impedance controller, we use simulations to validate the effectiveness and capabilities of our prioritized impedance controller based several control schemes.

### 3.5.1 Simulations for the Prioritized Weighted Optimization

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 P-WQP controller, we perform two simulations at the velocity  $1m.s^{-1}$ . Legged motion (e.g. walking) is not considered in this chapter but 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. Since the legged motion is only for active suspension which determines that the robot cannot turn. Therefore, the quadruped-CoM motion references for two simulation scenarios are set to be in the sagittal plane and its lateral reference is always zero.

In the first scenario, the robot runs on the flat terrain shown in Fig. 3.4(a). At the door axis, we install a virtual spring which generates a resistance force when the robot uses its manipulator to push it open, beginning at  $t \approx 7s$  (see Fig. 3.7(1c)). The end effector keeps in touch with the door for a while and changes its orientation according to the door motion. The second scenario relates to a more complex environment in which the robot runs on an uneven terrain (as can be seen in Fig. 3.4(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. 3.7(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. 3.7(2c).

The first simulation results are shown in Fig. 3.5 with index 1. The quadruped centroidal motion experiences a slight tracking fluctuation when the robot uses its manipulator to push the door. Along



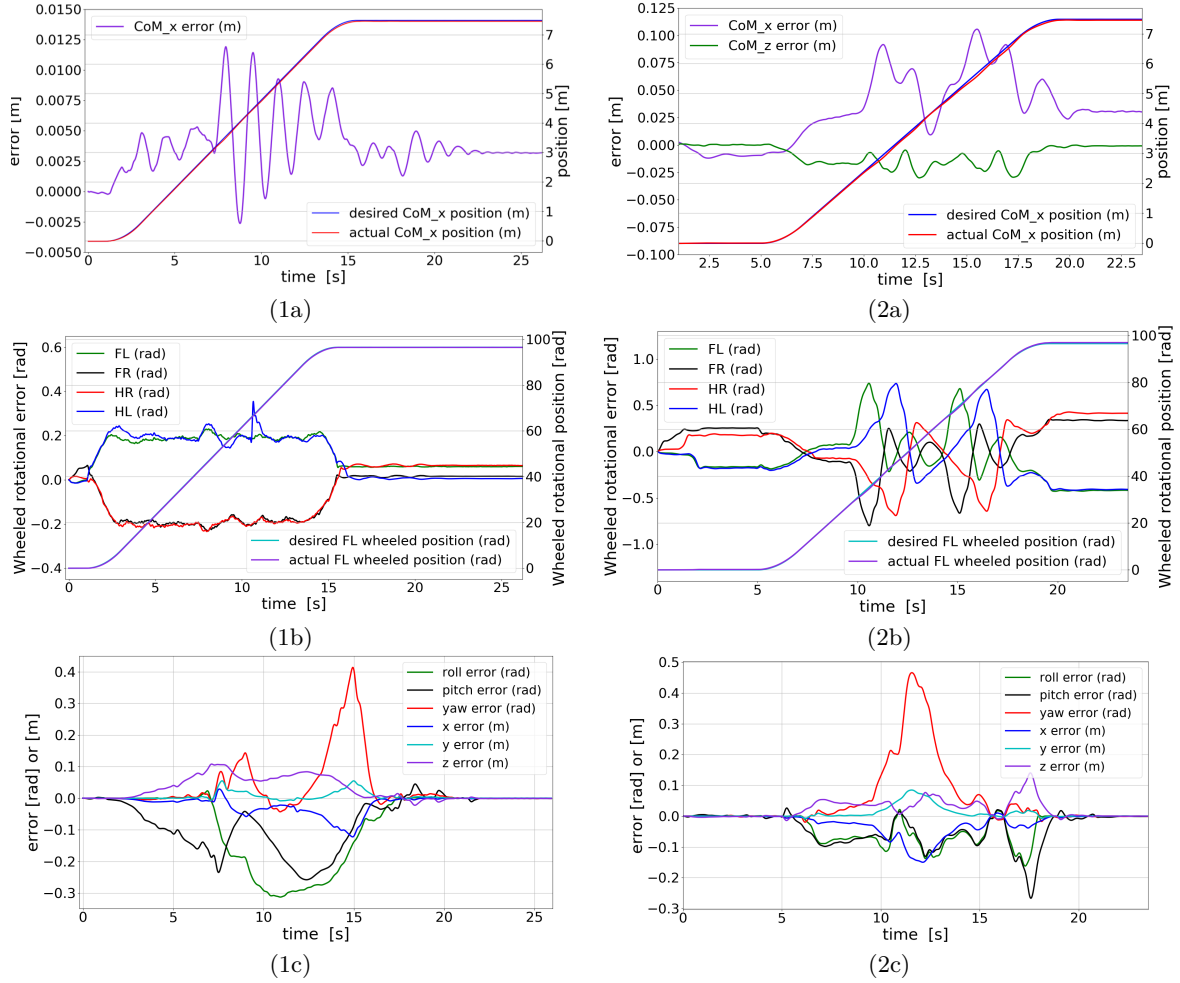


Figure 3.5: Plots of robot states for two simulation scenarios with index 1 and 2 respectively. (1a) Plot of quadruped CoM motion in forward  $x$  direction. (2a) Plot of quadruped CoM motion in forward  $x$  and vertical  $z$  direction. (1b) and (2b) Plot of the four wheels' rotational position errors: F, L, R, H represent front, left, right and hind. (1c) and (2c) Plot of the motion of the manipulator end effector: desired and actual orientations (roll, pitch, yaw) and positions ( $x$ ,  $y$ ,  $z$ ) of the end effector are relative to the base frame.

the whole process, the robot-CoM tracking error is very small, and the desired and actual trajectories are almost coincident as can be noticed in Fig. 3.5(1a). For the wheel motion, the four wheel tracking errors are plotted in Fig. 3.5(1b) and since four wheels experience the almost coincident trajectory, only the desired and actual front-left wheel positions are plotted in Fig. 3.5(1b). We can see that on flat terrain, even though the rolling friction is not unknown, by using our null-space based impedance controller in (3.5), the wheels can track the motion references with very small errors along the whole running process. For the manipulator motion, due to the unknown resistance force 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. 3.5(1c). The tracking errors for the three tasks can be adjusted by modifying the relative impedance controller parameters and the relative weight in the cost function. With our dynamics controller, the robot successfully finishes the missions with compliance.

The second simulation results for the tasks are also shown in Fig. 3.5 with index 2. The desired robot CoM position in vertical direction is  $0.4735m$ . Because of the existence of two bumps on the rough terrain, compared with the first simulation results, the CoM tracking in forward and vertical



directions experience a moderately larger error (see Fig. 3.5 (2a)), however, the errors are still very small along the whole process. We can see that the wheel tracking errors (see Fig. 3.5 (2b)) are larger than those provided when running on the flat ground. Since the rough geometry is unknown and the wheel-contact points are always estimated at the lowest points of the wheels in the wheel sagittal plane, it is not accurate when the robot crosses over the continuous bumps. The imprecise wheel-contact point estimation influences the outputs of the motion generator and the results of the quadruped-CoM tracking when crossing bumps, shown in Fig. 3.5(2a). This issue will be treated in our future works by developing an estimation method for the contact points. For the manipulator motion, to maintain the contact condition between the manipulator end effector and the wall, the manipulator trajectory is defined in a larger range. The end effector experiences a little higher error along the wall-washing motion, shown in Fig. 3.5(2c). However, even on unknown rough terrain and with large manipulator motion errors, by applying our compact dynamics controller and compatible impedance controllers, the robot successfully finish the missions with compliance and efficiency, especially tracking the quadruped-CoM motion.

### 3.5.2 Simulations for the Hybrid Hierarchical-weighted Optimization

In the scenario shown in Fig. 3.6, *TowrISIR* with the manipulator is applied. We also control three tasks, ranked by priority from high to low as the quadruped centroidal motion task, the wheel motion task and the manipulation task. The scenario in Fig. 3.6 relates to the robot driven on the rough terrain along while holding one handrail using its arm. After the robot leaves the handrail, the robot retracts its arm and squats down to cross through one desk (there is one bump under the desk). The positions of the handrail and the desk are known and the ground profile is unknown.

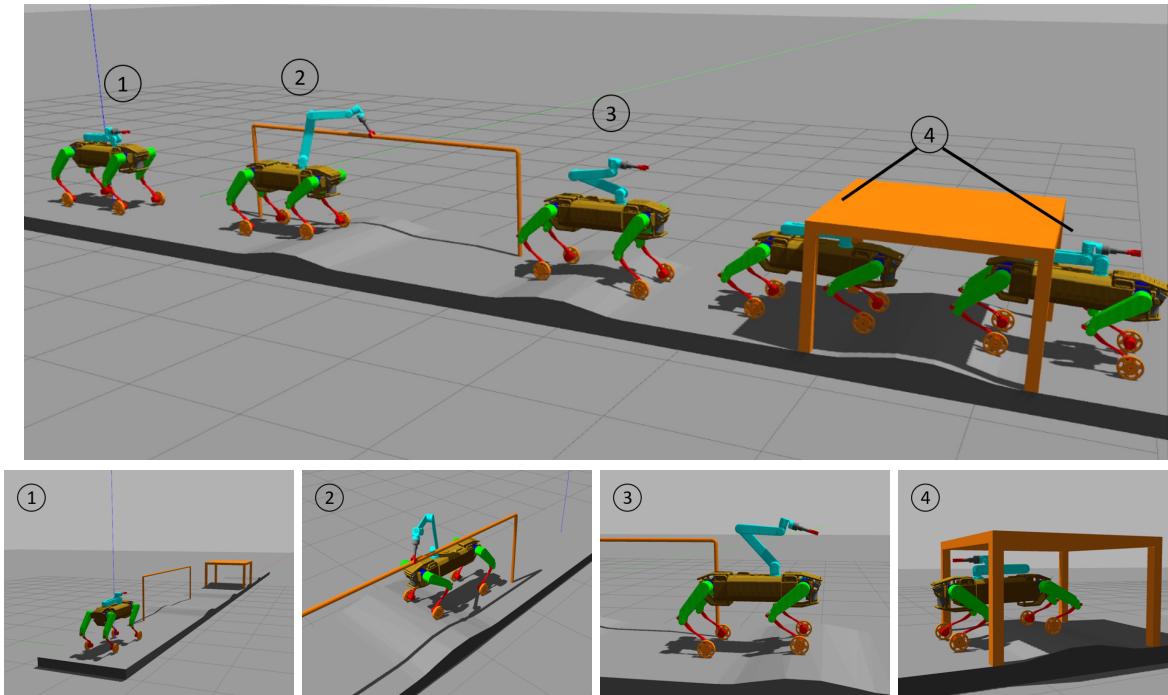


Figure 3.6: The robot is driven on various rough terrains while handling some activities. The whole process is divided into four stages. Stage 1: the beginning state. Stage 2: the robot holds the handrail using its arm. Stage 3: the robot retracts its arm after the handrail. Stage 4: the robot squats down and crosses through the desk and over one bump under the desk.

Firstly, we test the P-HWQP solver in Section 3.4.4, which combines the P-WQP solver in Section 3.4.2 and the P-HQP solver in Section 3.4.3. The centroidal motion task is artificially divided into 2

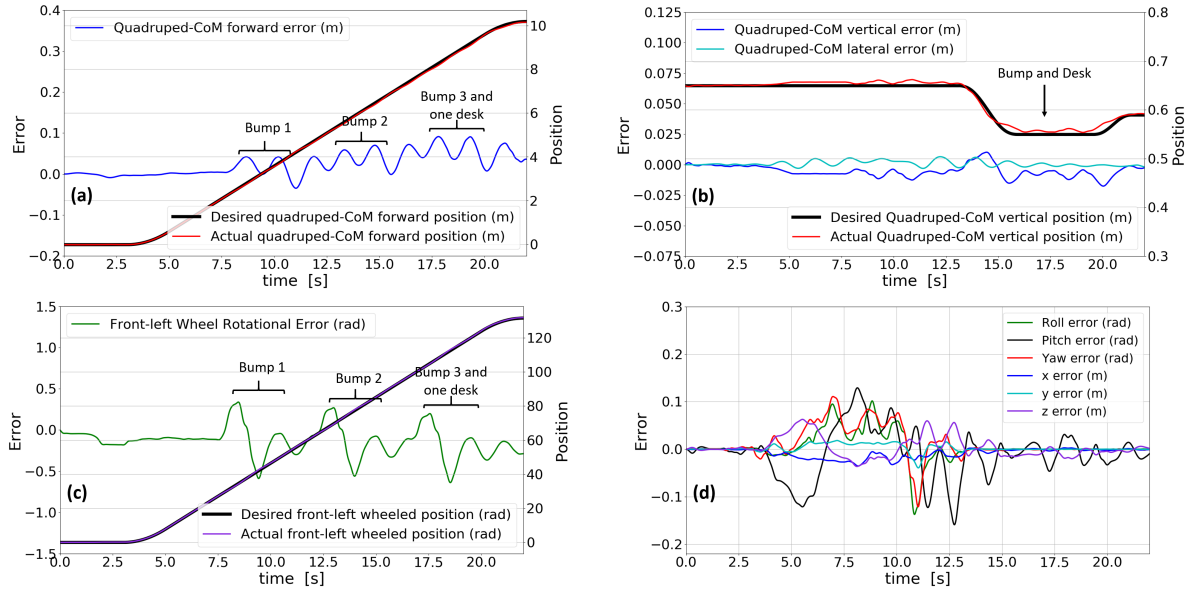


Figure 3.7: Plots of robot states by using the hybrid hierarchical-weighted control framework. The left axes denote the motion errors and the right axes represent positions. (a) plots the centroidal translational motion in the forward direction. (b) plots the centroidal translational motion in the vertical direction. (c) plots the front-left wheel rotational positions and the motion error. (d) plots the manipulator end effector motion errors, including the orientation errors (roll, pitch, yaw) and the position errors (x, y, z).

sub-tasks, including the translational motion sub-task and the angular motion sub-task, which enables to verify our hybrid control framework. For robots with more DOFs, e.g. humanoid robots, it is difficult to judge priorities among several tasks, therefore, our hybrid control scheme can be applied naturally. The simulation results are shown in Fig. 3.7. Along the whole process, the centroidal motion tracking errors are very small. The desired and actual trajectories are almost coincident as can be seen in Fig. 3.7(a) and 3.7(b). Even when the robot squats down to cross through one desk with an unknown bump under it, the prioritized impedance controller still enables the tracking performance of the CoM height reference with small errors. Since the quadruped-CoM translational motion is one sub-task in the null-space based weighted scheme, its tracking performance can also be improved by adjusting the weight in the weighted scheme. The legged motions satisfy the contact constraint and the task hierarchy, they follow the centroidal motion reference to achieve the squatting motion. The impedance controller for the legged motion indirectly influences the whole-body behaviors, and it is integrated into our prioritized impedance controller to guarantee the legged motion to be consistent with the task hierarchies and enable the robot to return back to its normal state after crossing over several bumps. The inaccurate contact model induces the wheel motion with moderately larger tracking errors (c.f. Fig. 3.7(c)) when the robot climbs on bumps. In addition, it is difficult to estimate the rolling resistance force due to the unknown rolling friction. However, the wheel motion error at the end of the whole process remains around  $0.15rad$  which is very small because of the component related to the wheel motion in the prioritized impedance controller. For the manipulator motion, its trajectory is defined in a bigger range to enable the robot to hold the handrail all the time. The orientation and position errors of the manipulator motion experience a relatively big fluctuation, c.f. Fig. 3.7(d). However, even the external touching-handrail force is unknown, the robot still successfully holds the handrail by using our hybrid dynamics controller integrated with our prioritized impedance forces. In this simulation scenario, even with several interaction uncertainties, including the unknown rough terrain with bumps, the imprecise contact points, the unknown rolling resistance and the unknown external force from the handrail, the robot successfully handles these missions with compliance, especially for the centroidal motion tracking mission, by using our compliant torque controller.

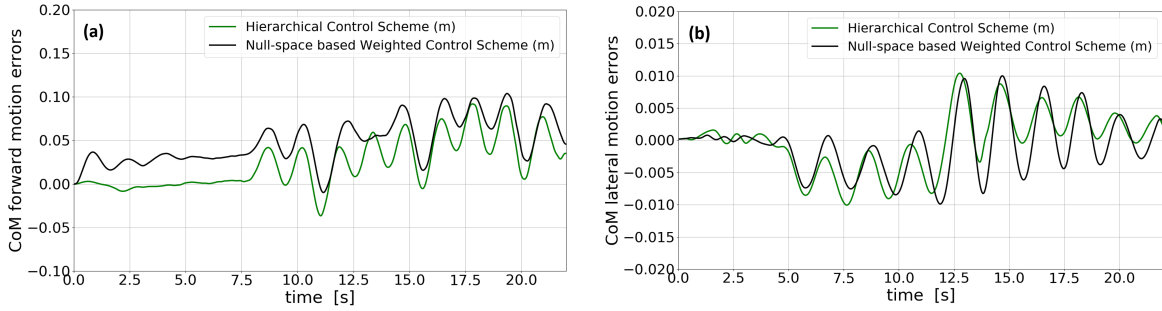


Figure 3.8: CoM tracking performance comparison by the hierarchical scheme and the null-space based weighted scheme. (a) plots the centroidal position errors in the forward direction. (b) plots the centroidal position errors in the lateral direction.

Then we simulate this scenario shown in Fig. 3.6 by using the P-WQP solver and the P-HQP solver respectively. They both can handle the missions successfully. We select to compare the performances relating to the CoM forward and lateral position errors by setting the same gains in the impedance controllers for the two solvers. For the weighted scheme, we artificially set the weight for the manipulator motion to be bigger than the CoM and wheel motions. The CoM tracking performances are shown in Fig. 3.8. We can see that the CoM tracking error is smaller in the forward direction using the hierarchical scheme than that using the weighted one, c.f. Fig. 3.8(a). In addition, the CoM lateral motion experiences a smaller fluctuation for the hierarchical solver (see Fig. 3.8(b)). The CoM tracking performance for the weighted scheme can be improved by adjusting the relative task weight. Even though the centroidal motion tracking has better performance by using P-HQP, the selection of the two solvers depends on the user-defined applications. If one application requires the strict hierarchy, and the tasks with low priorities cannot conflict the tasks with high priorities, P-HQP is a better option. If the tasks can only be distributed with importance levels without strict priorities, P-WQP is a better choice.

Finally, we use the same environment to compare the influences by one impedance controller on one specific task. We select to use the P-HQP solver for our quadruped-on-wheel robot, and the manipulator is not used to reduce influences from the external touching-handrail force to the tracking results of the centroidal and wheel motions. The centroidal motion task still has the high priority and the wheel motion task has the low priority. The gains for the impedance controller relating to the wheel motion in (3.5) are set with high values and low values. The other parameters are set to be all the same. Then the results for the two tasks are shown in Fig. 3.9. We can see that the wheel-joint motion with the low gain experiences an apparent larger fluctuation compared with that with the high gain, c.f. Fig 3.9(b). In contrast, the centroidal motion with high priority experiences very similar processes even the gains for the wheel motion are set to be quite different, c.f. Fig. 3.9(a). In this way, the hierarchical control between the two tasks is ensured.

In this section, all simulations are done on a usual PC with an i7-7700 CPU, 3.6GHz. The sampling time is fixed with 2ms. For the quadruped-on-wheel manipulator, one QP is solved at each time step which has 22 decision variables and 92 inequality constraints. The maximum calculation period is 1.2ms for the P-WQP solver, and is 1.35ms for the P-HQP and P-HWQP solvers, which is efficient and suitable for the real-time control of other locomotion robots with high DOFs. The selection of these torque controllers depend on the application requirements.

To summarize, the results provided by our approaches are quite satisfactory. The prioritized impedance controller based dynamics controllers enable the robot to handle complex tasks respecting contact constraints and hierarchy priorities. Even with the changing environment and unknown external forces, using our embedded prioritized impedance controller, our robot successfully handles the missions with compliance. For the three optimization based techniques, they can be selected depending on user-defined functions. When all tasks cannot be distributed with strict priorities, the null-space based weighted scheme can be selected. For applications with strict priorities, the hierarchical scheme is more

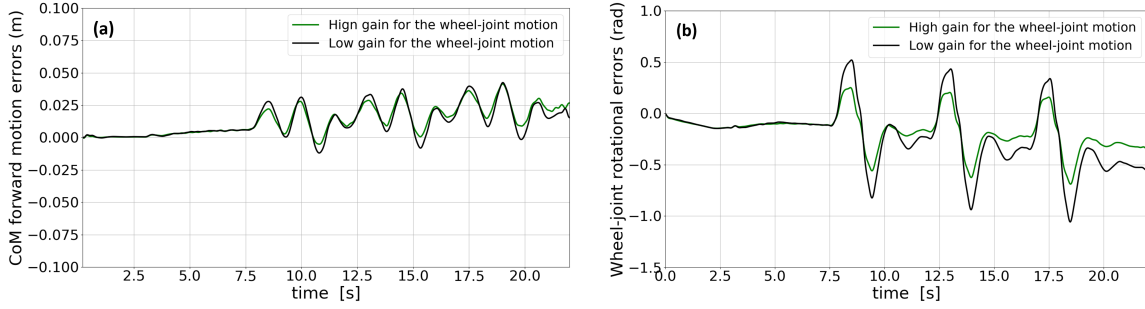


Figure 3.9: Hierarchical control between the centroidal motion tracking and the wheel motion tracking tasks. (a) plots the centroidal position errors in the forward direction. (b) plots the wheel-joint motion errors.

advantageous. The hybrid control framework is more general for applications in which some hierarchies only have one task in each one of them, and some hierarchies have several sub-tasks. The simulation results can also be referred in the video for this chapter. In the following chapters relating to the model based wheeled and legged motion generators, we will only introduce the differences in the relative torque control sections.

## 3.6 Summary

In this chapter, we propose the prioritized impedance controller. This strategy is integrated into four improved hierarchical torque control methods. In the pseudo-inverse dynamics approach, the generalized dynamics model is modified to integrate our prioritized impedance controller, which result in the multi-task torque control naturally with the hierarchical compliance. In the null-space weighted convex optimization method, the modified dynamics model is used to achieve the task acceleration extraction, in this way, the prioritized impedance controller is integrated. Compared with conventional weighted QP, the null-space based weighted scheme enables the impedance controllers to be decoupled to the specific designed sub-tasks, especially for impedance controllers which indirectly influence the designed tasks. In addition, for the torque controller based on the hierarchical convex optimization, we propose one new dynamics model to incorporate our prioritized impedance controller. To be emphasized, the new dynamics model embodies the multi-level control hierarchy, the hierarchical task reference is extracted from this model which enables various impedance controllers to be integrated compatibly with priorities. The inequality constraints are modified to be compatible with the optimization solver by extracting the quadratic forms of the contact force and the joint acceleration from our new dynamics model. Then we use our proposed model to integrate the prioritized impedance controller into a more complex scenario which combines the hierarchical and the null-space based weighted schemes together. The modified or new dynamics model enable the achievement of the more general and complete hierarchical optimization solvers, both for the task acceleration extraction and the same number of constraints. In the simulations, we test the pseudo-inverse dynamics and the optimization based techniques. The results show that the control schemes with the prioritized impedance controller enable our robot to handle complex missions efficiently and compliantly.

# Model-based Wheeled Locomotion

---

## Contents

---

<b>4.1 Introduction and Chapter Outline</b> . . . . .	<b>56</b>
<b>4.2 Wheel Motion Generator</b> . . . . .	<b>57</b>
4.2.1 Wheel Velocity Generator . . . . .	59
4.2.2 Wheel Acceleration Generator . . . . .	60
<b>4.3 Whole-body Motion Generator</b> . . . . .	<b>61</b>
4.3.1 Adaptive altitude Control Model . . . . .	62
4.3.2 Wheel-center Motion Relative to the Base . . . . .	63
4.3.3 Whole-body Velocity Generator . . . . .	64
4.3.4 Whole-body Acceleration Generator . . . . .	64
<b>4.4 Whole-body Torque Controller</b> . . . . .	<b>65</b>
4.4.1 Feedback Whole-body Impedance Controllers . . . . .	66
4.4.2 PIC based Torque Controller . . . . .	67
<b>4.5 Simulation on Various Rough Terrains</b> . . . . .	<b>68</b>
4.5.1 Running on Artificial Environment . . . . .	68
4.5.2 Running on 3D Rough Terrain . . . . .	70
4.5.3 Running on Terrain with Big Altitude Difference . . . . .	71
4.5.4 Running on Non-symmetric Geometry . . . . .	72
<b>4.6 Summary</b> . . . . .	<b>73</b>

---

This chapter proposes two kinds of motion generators for a quadruped-on-wheel robot which has the ability to cross various rough terrains with a much-altitude difference using the model-based whole-body torque control. The two motion generators both track the centroidal motion of one quadruped-on-wheel robot. The whole-body inverse kinematics model is derived using spatial vectors. In the first wheel motion generator, we assume that desired legged suspension is known, and the desired wheel motion can be achieved by combining the robot centroidal momentum & dynamics model and the whole-body kinematics model. In the second whole-body motion generator, the desired legged suspension can be derived on-line. The whole-body motion generator tracks the centroidal translational motion in the forward and lateral directions in the inertial frame, and the centroidal height motion relative to the base frame in the base vertical direction. Then the whole-body motion generator is developed by combining both the kinematics model, the wheel-center motion model, the robot centroidal momentum and dynamics models, and one proposed altitude control model. Finally, the robot is simulated to be driven on various rough terrains with compatible dynamic behaviors by using the developed torque controllers in Chapter 3. Both the feed-forward and feedback controllers respect the contact constraints to enable stable locomotion behaviors.

## 4.1 Introduction and Chapter Outline

From a practical point of view, quadruped-on-wheel robots combine both strengths of the two kinds of robots, and can perform more versatile locomotion modes from rolling to walking, running, jumping and

hybrid locomotion modes. However, these enhanced locomotion abilities come with more complex dynamics model, consistent whole-body motion generation and multi-task torque control. According to the state of the art relating to the wheel motion generation in Chapter Introduction, in this chapter, we also track the robot centroidal motion for our quadruped-on-wheel robot, and achieve the kinematics contact constraints in the second order. Then we step further and derive the relationship in the acceleration level between the centroidal motion and all wheel-leg motions. Therefore, the centroidal dynamics should be applied without the leg-massless assumption. Since the legged suspension results in the change of the whole-robot CoM relative to the base, we select to control this relative motion which is independent of the terrain geometry, and it enables the robot with one novel altitude control especially on unknown environment with a much-altitude difference.

In this chapter, we focus on the rolling based locomotion and propose a new adaptive whole-body 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. All wheels are in contact with the ground, and the legged motion is used as an active suspension and the base height control. In the first wheel motion generator, the desired legged suspension is pre-defined and the wheel motion generator is developed by combining both the kinematics model and the robot centroidal momentum/dynamics model and the models are decomposed into three components relating to the base motion, the legged motion and the wheel motion. Then we step further to develop the second whole-body motion generator by combining the whole-body inverse kinematics model, the wheel-center motion model, the robot centroidal momentum/dynamics model and one proposed altitude control model. Apart from the centroidal translational motion references in the forward and lateral directions of the inertial frame, we select to control the centroidal height motion relative to the base frame in the base vertical direction, which enables the robot to run on terrains with high roughness and high altitude variation. The whole-body velocity generator is established by using the actual robot configuration, and the whole-body acceleration generator is developed by applying the actual robot configuration and body velocities, which enable the robot with robustness and adaptiveness on unknown terrains. Finally, the torque controller P-HQP developed in Chapter 3 is used to track the centroidal motion and validate our whole-body motion generator.

The remainder of this chapter is organized as follows. Section 4.2 and Section 4.3 constitute the main contributions of this chapter which describes the new wheel-motion generator and the whole-body motion generator, which both track the robot centroidal motion. In the first wheel-motion generator, by given the desired the legged suspension, we achieve the wheel motion by combining the centroidal momentum & dynamics model and the whole-body kinematics model. In the second whole-body motion generator, the desired legged suspension is generated on-line, and we propose the altitude control model to interpret the robot internal motion for the robot height control. Then the desired whole-body motion is derived by combining the kinematics model, the altitude control model, the wheel-center motion model, and the centroidal momentum and dynamics models. In Section 4.4, the feedback impedance controller is presented and they are integrated into the torque controller in Chapter 3. In Section 4.5, we use the torque controller P-HQP to validate our whole-body motion generator through two numerical simulations using our virtual quadruped-on-wheel robot. Section 4.6 closes the chapter with the conclusion.

## 4.2 Wheel Motion Generator

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. 4.1, 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. 4.1, and the superscript  $d$  denotes the desired values of any vector. In this chapter, we concentrate on the rolling mode and all legs are maintained in contact with the ground. For



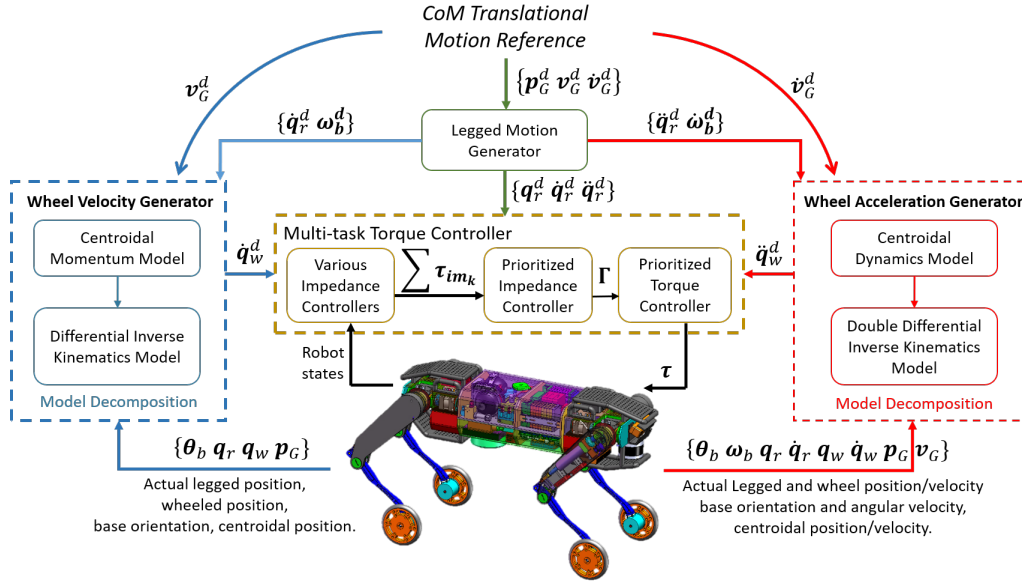


Figure 4.1: Wheel Motion Generation Framework: For the rolling locomotion mode, by given the quadruped-CoM translational motion reference, the legged locomotion generator outputs the desired legged configuration for the legged active suspension and the base height control. The superscript  $d$  means “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 4.2. 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 4.4.

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 4.4 enables the robot to handle various rough terrains successfully, shown in Section 4.5. Since the torque controller is built in the operational space, the desired legged motion setting is not used, detailed in Section 4.4.

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  $\{\mathbf{p}_G, \mathbf{v}_G^d, \boldsymbol{\theta}_b, \boldsymbol{\omega}_b^d, \mathbf{q}_r, \dot{\mathbf{q}}_r^d, \mathbf{q}_w\}$  and  $\{\mathbf{p}_G, \mathbf{v}_G, \dot{\mathbf{v}}_G^d, \boldsymbol{\theta}_b, \boldsymbol{\omega}_b, \dot{\boldsymbol{\omega}}_b^d, \mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r^d, \mathbf{q}_w, \dot{\mathbf{q}}_w\}$  respectively (see Fig. 4.1), and  $\mathbf{p}_G$  denotes the quadruped-CoM position. The global solving process: by given the inputs for the wheel velocity and acceleration generators, the base translational velocity  $\mathbf{v}_b$  and acceleration  $\dot{\mathbf{v}}_b$  can be extracted out from the centroidal momentum and dynamics models respectively. Then  $\mathbf{v}_b$  and  $\dot{\mathbf{v}}_b$  are substituted into the whole-body inverse kinematics models in (2.23) and (2.24) respectively. Then the wheel velocity  $\dot{\mathbf{q}}_w$  and acceleration  $\ddot{\mathbf{q}}_w$  can be derived. The desired wheel position can be estimated in each loop. Several algorithms listed in Tab. 4.1 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 4.4.1 which

is integrated into the prioritized impedance controller based hierarchical torque control framework in Chapter 3.

### 4.2.1 Wheel Velocity Generator

To track the robot centroidal translational velocity reference, the wheel motion  $\dot{\mathbf{q}}_w$  can be derived by combining the differential kinematic model in (4.1a) and the quadruped centroidal momentum model in (4.1b) as follows,

$$\mathbf{L}_w \dot{\mathbf{q}}_w = \mathbf{L} \begin{bmatrix} \mathbf{v}_b \\ \dot{\mathbf{q}}_r \end{bmatrix} = \mathbf{L}_{b_\omega} \boldsymbol{\omega}_b + \mathbf{L}_{b_v} \mathbf{v}_b + \mathbf{L}_r \dot{\mathbf{q}}_r, \quad (4.1a)$$

$$\mathbf{h}_G = \begin{bmatrix} \mathbf{k}_G \\ \mathbf{l}_G \end{bmatrix} = \mathbf{I}_G \mathbf{v}_G = \begin{bmatrix} \mathbf{I}_{G_\omega} & \mathbf{I}_{G_v} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_G \\ \mathbf{v}_G \end{bmatrix} = {}^b \mathbf{X}_G^T \sum_{j=1}^{N_B} {}^j \mathbf{X}_b^T \mathbf{I}_j \mathbf{v}_j, \quad (4.1b)$$

where (4.1a) is equivalent with (2.23b) by setting  $\mathbf{L} = [\mathbf{L}_{b_\omega} \quad \mathbf{L}_{b_v} \quad \mathbf{L}_r]$ , in which  $\mathbf{L}_{b_\omega}$  and  $\mathbf{L}_{b_v}$  are the first and second three columns of  $\mathbf{L}$  in (2.22) respectively.  $\mathbf{h}_G$  denotes the centroidal momentum and it is composed by the angular momentum  $\mathbf{k}_G$  and the translational momentum  $\mathbf{l}_G$ . In addition,  $\mathbf{v}_G$  represents the centroidal spatial velocity,  $\boldsymbol{\omega}_G$  and  $\mathbf{v}_G$  represent the centroidal angular and translational velocity respectively.  $\mathbf{I}_{G_\omega}$  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  $\mathbf{v}_j$  represent the body  $j$  spatial inertia and spatial velocity respectively. The body and joint indexes are shown in Fig. 2.3(a). 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 [Siciliano 2016]. Therefore, we have the relationship between  $\ddot{\mathbf{p}}_G$  and  $\dot{\mathbf{v}}_G$  as follows,

$$\mathbf{v}_G = \dot{\mathbf{p}}_G, \quad (4.2a)$$

$$\dot{\mathbf{v}}_G = \ddot{\mathbf{p}}_G - \boldsymbol{\omega}_G \times \mathbf{v}_G, \quad (4.2b)$$

and the relationship between  $\ddot{\mathbf{p}}_b$  and  $\dot{\mathbf{v}}_b$  is written in (3.38).

The centroidal momentum model in (4.1b) can be decomposed into components relating to the base motion, the legged motion and the wheeled motion as follows,

$$\mathbf{I}_G \begin{bmatrix} \boldsymbol{\omega}_G \\ \mathbf{v}_G \end{bmatrix} = \mathbf{Q}_b \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_b \end{bmatrix} + \mathbf{Q}_r \dot{\mathbf{q}}_r + \mathbf{Q}_w \dot{\mathbf{q}}_w, \quad (4.3)$$

where  $\mathbf{Q}_r \dot{\mathbf{q}}_r \in \mathbb{R}^6$  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. 4.1(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,

$$\mathbf{Q}_b = [\mathbf{Q}_{b_\omega} \quad \mathbf{Q}_{b_v}] = \mathbf{I}_G \mathbf{J}_G^b = {}^b \mathbf{X}_G^T \mathbf{I}_b^C, \quad (4.4a)$$

$$\mathbf{Q}_w = {}^b \mathbf{X}_G^T [{}^{w_1} \mathbf{X}_b^T \mathbf{I}_{w_1} \boldsymbol{\Phi}_{w_1} \quad \cdots \quad {}^{w_4} \mathbf{X}_b^T \mathbf{I}_{w_4} \boldsymbol{\Phi}_{w_4}], \quad (4.4b)$$

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 (4.3) as follows,

$$\mathbf{v}_b = \boldsymbol{\Gamma}_{Gb} \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_G \end{bmatrix} - \boldsymbol{\Gamma}_r \dot{\mathbf{q}}_r - \boldsymbol{\Gamma}_w \dot{\mathbf{q}}_w, \quad (4.5)$$

where  $\boldsymbol{\Gamma}_{Gb}$ ,  $\boldsymbol{\Gamma}_r \dot{\mathbf{q}}_r$  and  $\boldsymbol{\Gamma}_w$  are derived as below,

$$\boldsymbol{\Gamma}_{Gb} = \mathbf{S}_c [-\mathbf{I}_{G_\omega} \quad \mathbf{Q}_{b_v}]^{-1} [-\mathbf{Q}_{b_\omega} \quad \mathbf{I}_{G_v}] \in \mathbb{R}^{3 \times 6}, \quad (4.6a)$$

$$\boldsymbol{\Gamma}_r \dot{\mathbf{q}}_r = \mathbf{S}_c [-\mathbf{I}_{G_\omega} \quad \mathbf{Q}_{b_v}]^{-1} \mathbf{Q}_r \dot{\mathbf{q}}_r \in \mathbb{R}^3, \quad (4.6b)$$

$$\boldsymbol{\Gamma}_w = \mathbf{S}_c [-\mathbf{I}_{G_\omega} \quad \mathbf{Q}_{b_v}]^{-1} \mathbf{Q}_w \in \mathbb{R}^{3 \times 4}. \quad (4.6c)$$



Table 4.1: Algorithms for the Wheel Motion Generation

---

(a) Algorithm for  $\mathbf{Q}_r \dot{\mathbf{q}}_r$  in (4.3) 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**

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

---

(b) Algorithm for  $\mathbf{Q}_r \ddot{\mathbf{q}}_r$  and  $\mathbf{Q}_{bias}$  in (4.10) 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**

$\mathbf{Q}_r \ddot{\mathbf{q}}_r = {}^1\mathbf{X}_G^T \mathbf{E}$  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 achieve the relationship between the base translational velocity and the wheel velocity in (4.5), and it is substituted into the differential kinematics model in (4.1a),

$$(\mathbf{L}_w + \mathbf{L}_{b_w} \Gamma_w) \dot{\mathbf{q}}_w = \mathbf{L}_{b_w} \left( \Gamma_{Gb} \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_G \end{bmatrix} - \Gamma_r \dot{\mathbf{q}}_r \right) + \mathbf{L}_r \dot{\mathbf{q}}_r + \mathbf{L}_{b_w} \boldsymbol{\omega}_b, \quad (4.7)$$

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

After achieving the wheel velocity  $\dot{\mathbf{q}}_w$ , 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,

$$\begin{bmatrix} \boldsymbol{\omega}_G \\ \mathbf{v}_b \end{bmatrix} = [\mathbf{I}_{G_w} \quad -\mathbf{Q}_{b_w}]^{-1} \left( [\mathbf{Q}_{b_w} \quad -\mathbf{I}_{G_v}] \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_G \end{bmatrix} + \mathbf{Q}_r \dot{\mathbf{q}}_r + \mathbf{Q}_w \dot{\mathbf{q}}_w \right). \quad (4.8)$$

Then all the velocity references will be used for the whole-body compliance control in Section 3.3.

## 4.2.2 Wheel Acceleration Generator

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

$$\mathbf{L}_w \ddot{\mathbf{q}}_w = -\mathbf{L}_{b_w} \dot{\boldsymbol{\omega}}_b - \mathbf{L}_{b_w} \dot{\mathbf{v}}_b - \mathbf{L}_d, \quad (4.9a)$$

$$\dot{\mathbf{h}}_G = \begin{bmatrix} \dot{\mathbf{k}}_G \\ \dot{\mathbf{l}}_G \end{bmatrix} = \mathbf{I}_G \dot{\mathbf{v}}_G + \mathbf{v}_G \times \mathbf{I}_G \mathbf{v}_G = {}^b\mathbf{X}_G^T \sum_{j=1}^{N_B} {}^j\mathbf{X}_b^T (\mathbf{I}_j \mathbf{a}_j + \mathbf{v}_j \times \mathbf{I}_j \mathbf{v}_j), \quad (4.9b)$$

where  $\mathbf{a}_j$  is the body  $j$  spatial acceleration. (4.9a) is equivalent with (2.24)  $\mathbf{L}_d = \mathbf{L}_r \ddot{\mathbf{q}}_r + \dot{\mathbf{J}}_c \dot{\mathbf{q}}$ .

Then we decompose the centroidal dynamics model in (4.9b) into components relating to the base motion, the legged motion, the wheel motion and one bias item as below,

$$\mathbf{I}_G \begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \dot{\mathbf{v}}_G \end{bmatrix} = \mathbf{Q}_b \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_b \end{bmatrix} + \mathbf{Q}_r \ddot{\mathbf{q}}_r + \mathbf{Q}_w \ddot{\mathbf{q}}_w - \mathbf{Q}_{bias}, \quad (4.10)$$

where  $\mathbf{Q}_r \ddot{\mathbf{q}}_r \in \mathbb{R}^6$  and  $\mathbf{Q}_{bias} \in \mathbb{R}^6$  can be calculated recursively using the algorithm in Tab. 4.1(d).

To achieve the wheel accelerations, firstly we extract the base translational acceleration out from (4.10) as below,

$$\dot{\mathbf{v}}_b = \boldsymbol{\Gamma}_{Gb} \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} - \boldsymbol{\Gamma}_w \ddot{\mathbf{q}}_w + \boldsymbol{\Gamma}_d, \quad (4.11)$$

where  $\boldsymbol{\Gamma}_{Gb}$  and  $\boldsymbol{\Gamma}_w$  are referred to (4.6), and  $\boldsymbol{\Gamma}_d \in \mathbb{R}^3$  is derived as follows,

$$\boldsymbol{\Gamma}_d = \boldsymbol{\Gamma}_{bias} - \boldsymbol{\Gamma}_r \ddot{\mathbf{q}}_r, \quad (4.12a)$$

$$\boldsymbol{\Gamma}_{bias} = \mathbf{S}_c [-\mathbf{I}_{G\omega} \quad \mathbf{Q}_{b\omega}]^{-1} \mathbf{Q}_{bias}, \quad (4.12b)$$

$$\boldsymbol{\Gamma}_r \ddot{\mathbf{q}}_r = \mathbf{S}_c [-\mathbf{I}_{G\omega} \quad \mathbf{Q}_{b\omega}]^{-1} \mathbf{Q}_r \ddot{\mathbf{q}}_r. \quad (4.12c)$$

In this way, we have achieved the relationship between the base translational acceleration and the wheel acceleration in (4.11). Then we substitute this relationship function (4.11) into the double-differential kinematics model (4.9a),

$$(\mathbf{L}_w - \mathbf{L}_{b_v} \boldsymbol{\Gamma}_w) \ddot{\mathbf{q}}_w = -\mathbf{L}_{b_v} \left( \boldsymbol{\Gamma}_{Gb} \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} + \boldsymbol{\Gamma}_d \right) - \mathbf{L}_d - \mathbf{L}_{b\omega} \dot{\boldsymbol{\omega}}_b, \quad (4.13)$$

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,

$$\begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \dot{\mathbf{v}}_b \end{bmatrix} = [\mathbf{I}_{G\omega} \quad -\mathbf{Q}_{b\omega}]^{-1} \left( [\mathbf{Q}_{b\omega} \quad -\mathbf{I}_{Gv}] \begin{bmatrix} \dot{\boldsymbol{\omega}}_b \\ \dot{\mathbf{v}}_G \end{bmatrix} + \mathbf{Q}_r \ddot{\mathbf{q}}_r + \mathbf{Q}_w \ddot{\mathbf{q}}_w - \mathbf{Q}_{bias} \right). \quad (4.14)$$

In addition, the wheel position can be estimated in each control loop as follows,

$$\mathbf{q}_{w|k+1} = \mathbf{q}_{w|k} + \delta t \dot{\mathbf{q}}_{w|k} + \frac{1}{2} \delta t^2 \ddot{\mathbf{q}}_{w|k}, \quad (4.15)$$

where  $\mathbf{q}_{w|k}$  applies the actual wheel position in the  $k^{th}$  control loop. The  $k^{th}$ -loop velocity and acceleration apply the generated desired values. The  $(k+1)^{th}$ -loop desired wheel position is used in the impedance controller in Section 4.4.1.

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. However, we assume that the desired legged motion are constant in this section which is not suitable for more complex motion, e.g. squatting-down motion. Therefore, in the next section, we develop one general wheel motion generator.

### 4.3 Whole-body Motion Generator

When the robot runs on rough terrains, tracking the robot 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 in Fig. 4.2, the centroidal

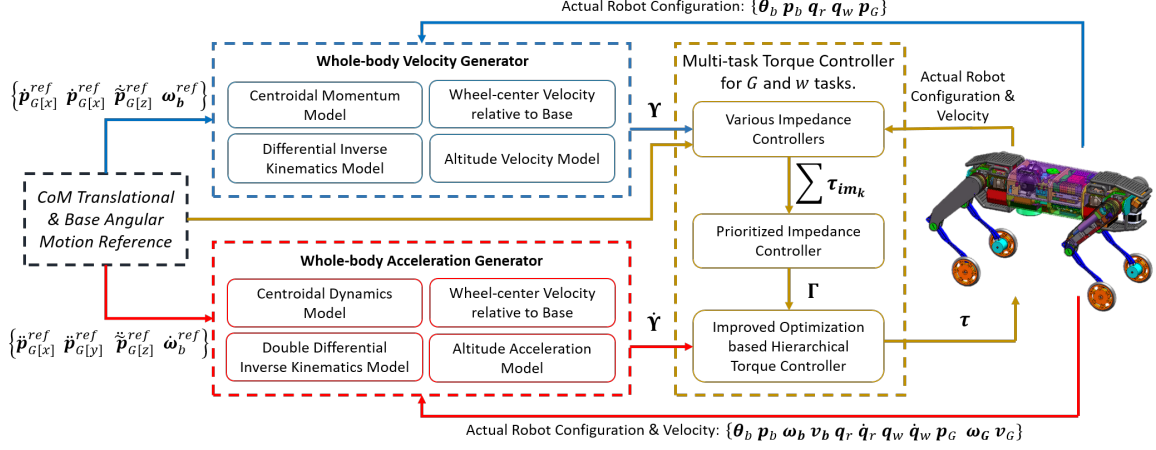


Figure 4.2: Whole-body Motion Generation Framework: For the rolling locomotion mode, by given the quadruped-CoM translational motion reference. The superscript *ref* means the reference. The whole-body velocity and acceleration are derived using the velocity and acceleration based models in the dashed blue and red boxes, detailed in Section 4.3. The outputs of the generators ( $\Upsilon$ ,  $\dot{\Upsilon}$ ), the desired whole-body motions, serve as the inputs of the prioritized impedance controller based torque controller in the yellow dashed box, and  $G$  and  $w$  denote the centroidal motion and the wheel motion tasks.

translational motion and the base angular motion serve as the references. Its forward motion is achieved by the wheeled locomotion, and its vertical and lateral motion are handled by the legged suspension, e.g. the squatting down motion. The base angular motion reference is also set with zero vectors,  $\omega_b^{ref} = \dot{\omega}_b^{ref} = \mathbf{0}_{3 \times 1}$ . The centroidal motion references include two parts, one relates to the centroidal translational motion in the forward and lateral directions of the inertial frame ( $\mathbf{p}_{G[x]}^{ref}$ ,  $\mathbf{p}_{G[y]}^{ref}$ ), and one relates to the centroidal translational motion w.r.t the base frame  $\tilde{\mathbf{p}}_{G[z]}^{ref}$ , shown in Fig. 4.2 and Fig. 4.3.

We propose one new whole-body motion generator by combining one novel altitude control model, the wheel-center motion model, and the centroidal momentum and dynamics model. By given the motion references, these models are combined with the inverse kinematics models in Section 2.1.3 to achieve the whole-body velocities and accelerations, shown in Fig. 4.2. For whole-body velocity generator, the robot actual configuration is used. For the whole-body acceleration generator, the robot actual configuration and body velocities are applied, which enables the robot to adapt effectively on unknown environment. The outputs of the whole-body velocity generator are used in the impedance controllers in Section 3.3, and the outputs of the whole-body acceleration generator are used as the feed-forward torque controller in Section 4.4.2.

### 4.3.1 Adaptive altitude Control Model

Without leg-massless assumption, the change of the legged configuration leads to the change of the robot CoM position relative to the base. To enable the robot to run on different-altitude geometry, we select to control the CoM height relative to the base frame along its  $z_b$  vertical direction which interprets the robot internal motion. This relative motion is independent of the terrain geometry, and this strategy avoids the complex trajectory generation of the base motion or the CoM motion in the vertical direction. The relationships between the centroidal translational motion in the base frame and in the inertial frame are achieved as follows,

$$\dot{\mathbf{p}}_G = {}^b \mathbf{R}_G^T \dot{\tilde{\mathbf{p}}}_G + \dot{\mathbf{p}}_b - {}^b \mathbf{R}_G^T \tilde{\mathbf{p}}_G \times \omega_b, \quad (4.16a)$$

$$\ddot{\mathbf{p}}_G = {}^b \mathbf{R}_G^T \ddot{\tilde{\mathbf{p}}}_G + \ddot{\mathbf{p}}_b - {}^b \mathbf{R}_G^T \tilde{\mathbf{p}}_G \times \dot{\omega}_b + \kappa_a, \quad (4.16b)$$

where  ${}^b \mathbf{R}_G$  transforms 3D vectors from the CoM frame to the base frame and  $({}^b \dot{\mathbf{R}}_G)^T = {}^b \mathbf{R}_G^T \omega_b \times$ . In addition,  $\mathbf{p}_G$  denotes the quadruped-CoM position in the inertial frame.  $\tilde{\mathbf{p}}_G$  denotes the robot CoM

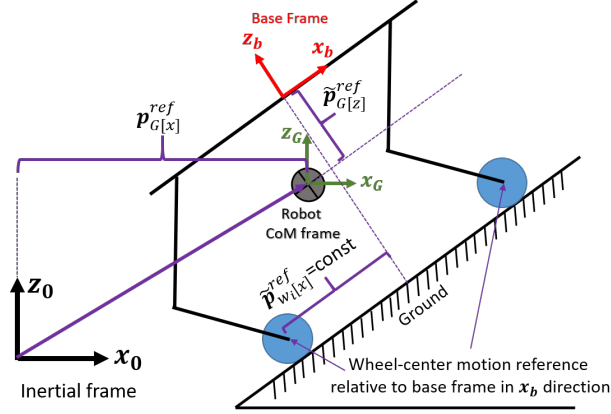


Figure 4.3: The motion references are shown in the robot sagittal plane.  $\mathbf{p}_{G[x]}^{ref}$  denotes the centroidal translational motion in the forward direction of the inertial frame.  $\tilde{\mathbf{p}}_{G[z]}^{ref}$  represents the centroidal translational motion relative to the base frame in  $z_b$  direction.  $\tilde{\mathbf{p}}_{w_i[x]}^{ref}$  is the leg  $i$  wheel-center motion reference relative to the base frame in  $x_b$  direction.

position in the base frame.  $\boldsymbol{\kappa}_a$  depicts the Coriolis and centripetal accelerations and it depends on the body velocities and it is achieved as below,

$$\boldsymbol{\kappa}_a = 2 {}^b \mathbf{R}_G^T \boldsymbol{\omega}_b \times \dot{\tilde{\mathbf{p}}}_G - {}^b \mathbf{R}_G^T \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_G \times \boldsymbol{\omega}_b. \quad (4.17a)$$

Therefore, by combining the relationships in (4.2) and (3.38), the centroidal translational motion relative to the base frame in  $z_b$  direction can be achieved as follows,

$$\mathbf{S}_z \dot{\tilde{\mathbf{p}}}_G = \dot{\tilde{\mathbf{p}}}_{G[z]} = \mathbf{S}_z ({}^b \mathbf{R}_G \mathbf{v}_G - \mathbf{v}_b + \tilde{\mathbf{p}}_G \times \boldsymbol{\omega}_b), \quad (4.18a)$$

$$\mathbf{S}_z \ddot{\tilde{\mathbf{p}}}_G = \ddot{\tilde{\mathbf{p}}}_{G[z]} = \mathbf{S}_z ({}^b \mathbf{R}_G \dot{\mathbf{v}}_G - \dot{\mathbf{v}}_b + \tilde{\mathbf{p}}_G \times \dot{\boldsymbol{\omega}}_b) + \boldsymbol{\kappa}_b, \quad (4.18b)$$

where  $\mathbf{S}_z = [0 \ 0 \ 1]$  is used to select the third component of any 3D vector.  $\mathbf{A}_b$  depends on the body positions and velocities as,

$$\boldsymbol{\kappa}_b = \mathbf{S}_z [{}^b \mathbf{R}_G (\boldsymbol{\omega}_G \times \mathbf{v}_G) - \boldsymbol{\omega}_b \times \mathbf{v}_b - {}^b \mathbf{R}_G \boldsymbol{\kappa}_a]. \quad (4.19)$$

It should be emphasized that in this section, the actual robot configuration is applied in the differential kinematics equations, and the actual robot configuration/velocity is used in the double-differential kinematics derivations. (4.18a) and (4.18b) serve as the constraints to control the centroidal motion relative to the base frame along  $z_b$  direction, which enables the robot to handle terrains with high altitude variation.

### 4.3.2 Wheel-center Motion Relative to the Base

We simply set the desired wheel-center position  $\tilde{\mathbf{p}}_{w_i} \in \mathbb{R}^3$  to be constant relative to the base frame in  $x_b$  forward direction. This means that  $\tilde{\mathbf{p}}_{w_i[x]}^{ref}$  is constant, as shown in Fig. 4.3. Therefore, we have the following kinematics relationships in the base forward and lateral directions as follows,

$$\mathbf{S}_x \dot{\tilde{\mathbf{p}}}_{w_i} = \mathbf{S}_x \tilde{\mathbf{J}}_{w_i} \dot{\mathbf{q}}_{r_i} = 0, \quad (4.20a)$$

$$\mathbf{S}_x \ddot{\tilde{\mathbf{p}}}_{w_i} = \mathbf{S}_x \tilde{\mathbf{J}}_{w_i} \ddot{\mathbf{q}}_{r_i} + \mathbf{S}_x \dot{\tilde{\mathbf{J}}}_{w_i} \dot{\mathbf{q}}_{r_i} = 0, \quad (4.20b)$$

where  $\mathbf{S}_x = [1 \ 0 \ 0]$  is used to select the first component of any 3-dimensional vector.  $\tilde{\mathbf{J}}_{w_i} \in \mathbb{R}^{3 \times 3}$  is the Jacobian relating to the wheel-center velocity relative to the base. The four wheel-center motions can

be combined as below,

$$\tilde{\mathbf{J}}_w \dot{\mathbf{q}}_r = \mathbf{0}_{n_r \times 1}, \quad (4.21a)$$

$$\tilde{\mathbf{J}}_w \ddot{\mathbf{q}}_r = \mathbf{A}_w, \quad (4.21b)$$

where  $\tilde{\mathbf{J}}_w \in \mathbb{R}^{n_w \times n_r}$  and  $\mathbf{A}_w \in \mathbb{R}^{n_w \times 1}$  are expressed as follows,

$$\tilde{\mathbf{J}}_w = \begin{bmatrix} \mathbf{S}_x \tilde{\mathbf{J}}_{w_1} \\ \vdots \\ \mathbf{S}_x \tilde{\mathbf{J}}_{w_4} \end{bmatrix}, \quad \mathbf{A}_w = \begin{bmatrix} -\mathbf{S}_x \dot{\tilde{\mathbf{J}}}_{w_1} \dot{\mathbf{q}}_{r_1} \\ \vdots \\ -\mathbf{S}_x \dot{\tilde{\mathbf{J}}}_{w_4} \dot{\mathbf{q}}_{r_4} \end{bmatrix}, \quad (4.22)$$

In this way, (4.21a) and (4.21b) serve as the extra constraints for the whole-body velocity and acceleration generators in Section 4.2.1 and 4.2.2 respectively.

### 4.3.3 Whole-body Velocity Generator

The desired whole-body velocity can be achieved by combining the differential kinematic model in (2.23b), the altitude control model in the velocity level in (4.18a), the wheel-center velocity constraint in (4.21a), and the quadruped centroidal momentum model in (4.3).  $\mathbf{Q}_r \in \mathbb{R}^{6 \times n_r}$  and  $\mathbf{Q}_w \in \mathbb{R}^{6 \times 4}$  are derived in another way directly as below,

$$\mathbf{Q}_r = \mathbf{I}_G \begin{bmatrix} \mathbf{J}_G^{r_1} & \mathbf{J}_G^{r_2} & \mathbf{J}_G^{r_3} & \mathbf{J}_G^{r_4} \end{bmatrix} \in \mathbb{R}^{6 \times n_r}, \quad (4.23a)$$

$$\mathbf{Q}_w = \mathbf{I}_G \begin{bmatrix} \mathbf{J}_G^{w_1} & \mathbf{J}_G^{w_2} & \mathbf{J}_G^{w_3} & \mathbf{J}_G^{w_4} \end{bmatrix} \in \mathbb{R}^{6 \times n_w}, \quad (4.23b)$$

where  $\mathbf{Q}_{b_w}$  and  $\mathbf{Q}_{b_v}$  are the left and right three columns of  $\mathbf{Q}_b$ . Then we combine (4.3), (4.18a), (4.21a) and (2.23b) together to achieve the whole-body motion references in the velocity level as follows,

$$\mathbf{\Omega} \Upsilon = \begin{bmatrix} \mathbf{0}_{1 \times n} & \dot{\mathbf{p}}_{G[x]}^{ref} & \dot{\mathbf{p}}_{G[y]}^{ref} & \dot{\mathbf{p}}_{G[z]}^{ref} & (\boldsymbol{\omega}_b^{ref})^T \end{bmatrix}^T, \quad (4.24)$$

where the superscript *ref* denotes the reference variables.  $\Upsilon \in \mathbb{R}^{n+6}$  denotes the calculated desired whole-body velocity, and  $\mathbf{\Omega}$  can be expressed as below,

$$\mathbf{\Omega} = \begin{bmatrix} (\boldsymbol{\omega}_G^d)^T & (\mathbf{v}_G^d)^T & (\boldsymbol{\omega}_b^d)^T & (\mathbf{v}_b^d)^T & (\dot{\mathbf{q}}_r^d)^T & (\dot{\mathbf{q}}_w^d)^T \end{bmatrix}^T, \quad (4.25)$$

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{I}_{G_w} & \mathbf{I}_{G_v} & -\mathbf{Q}_{b_w} & -\mathbf{Q}_{b_v} & -\mathbf{Q}_r & -\mathbf{Q}_w \\ \mathbf{0}_{n_w \times 3} & \mathbf{0}_{n_w \times 3} & \mathbf{0}_{n_w \times 3} & \mathbf{0}_{n_w \times 3} & \tilde{\mathbf{J}}_w & \mathbf{0}_{n_w \times n_w} \\ \mathbf{0}_{3n_c \times 3} & \mathbf{0}_{3n_c \times 3} & \mathbf{L}_{b_w} & \mathbf{L}_{b_v} & \mathbf{L}_r & \mathbf{L}_w \\ \mathbf{0}_{2 \times 3} & \mathbf{S}_{xy} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times n_r} & \mathbf{0}_{2 \times n_w} \\ \mathbf{0}_{1 \times 3} & \mathbf{S}_z^b \mathbf{R}_G & \mathbf{S}_z \tilde{\mathbf{p}}_G \times & -\mathbf{S}_z & \mathbf{0}_{1 \times n_r} & \mathbf{0}_{1 \times n_w} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times n_r} & \mathbf{0}_{3 \times n_w} \end{bmatrix}, \quad (4.26)$$

where all the superscript *d* means the generated desired variables.  $\mathbf{S}_{xy} = [\mathbf{1}_{2 \times 2} \ \mathbf{0}_{2 \times 1}]$ . In  $\mathbf{\Omega}$ , the top three lines relate to the centroidal momentum model in (4.3), the wheel-center velocity in (4.21a) and the contact constraint in (2.23b) respectively. The fourth line denotes the centroidal velocity reference in the forward and lateral directions by multiplying  $\mathbf{S}_{xy}$  on both sides of (4.2a). The fifth line represents the centroidal velocity reference relative to the base in  $z_b$  direction, detailed in (4.18a). The last line relates to the base angular velocity reference. In this subsection, we derive the desired whole-body motion in the velocity level  $\Upsilon$  and it will be used for the whole-body compliance controllers in Section 4.4.

### 4.3.4 Whole-body Acceleration Generator

The desired whole-body acceleration can be derived using the double-differential kinematic model in (2.24c), the altitude control model in the acceleration level in (4.18b), the wheel-center acceleration

constraint in (4.21b), and the centroidal dynamics model in (4.10).  $\mathbf{Q}_{bias} \in \mathbb{R}^6$  depends on body velocities and can also be calculated directly as below,

$$\mathbf{Q}_{bias} = {}^b\mathbf{X}_G^T \mathbf{U} \mathbf{C}, \quad (4.27)$$

Then we combine (4.10), (4.18b), (4.21b) and (2.24c) together to achieve the desired whole-body motion in the acceleration level as follows,

$$\Omega \dot{\mathbf{Y}} = \begin{bmatrix} \mathbf{B}_d^T & \mathbf{A}_w^T & \mathbf{B}_c^T & \mathbf{B}_r^T & (\ddot{\mathbf{p}}_{G[z]}^{ref} - \boldsymbol{\kappa}_b) & (\dot{\boldsymbol{\omega}}_b^{ref})^T \end{bmatrix}^T, \quad (4.28)$$

where  $\mathbf{B}_c$ ,  $\mathbf{B}_d$  and  $\mathbf{B}_r$  are achieved as follows,

$$\mathbf{B}_c = -\mathbf{J}_c \dot{\mathbf{q}}, \quad (4.29a)$$

$$\mathbf{B}_d = \mathbf{Q}_{bias} - \mathbf{v}_G \times \mathbf{I}_G \mathbf{v}_G, \quad (4.29b)$$

$$\mathbf{B}_r = \begin{bmatrix} \dot{\mathbf{v}}_{G[x]}^{ref} \\ \dot{\mathbf{v}}_{G[y]}^{ref} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{p}}_{G[x]}^{ref} \\ \dot{\mathbf{p}}_{G[y]}^{ref} \end{bmatrix} - \mathbf{S}_{xy}(\boldsymbol{\omega}_G \times \mathbf{v}_G). \quad (4.29c)$$

In (4.28), the top three lines relate to the centroidal dynamics model in (4.10), the wheel-center acceleration in (4.21b) and the derivative contact constraint in (2.24c) respectively. The fourth line represents the centroidal acceleration reference in the forward and lateral directions. The fifth line denotes the centroidal acceleration reference relative to the base along  $z_b$  direction, detailed in (4.18b). The last line relates to the base angular acceleration reference. Then the desired whole-body motion in the acceleration level  $\dot{\mathbf{Y}}$  will be used in the PIC based torque controller in Chapter 3. In addition, the desired wheel position can also be estimated in each control loop using (4.15). It is noticed that the joint position  $\mathbf{q}_r$  is not required in the torque controller.

In this section, we achieve the desired whole-body motion in the velocity and acceleration levels by combining the whole-body inverse kinematics model, the altitude control model, the wheel-center motion model, and the centroidal momentum and dynamics models. This method can be used for other locomotion robots with wheels. The altitude control model is integrated in the whole-body motion generators, and enables the robot to run on the different-altitude geometry. The actual robot configuration is applied to achieve the desired whole-body velocity, and the actual robot configuration/velocity is used to achieve the desired whole-body acceleration, which enables the robot with robustness and adaptiveness on various unknown terrains. In addition, the wheel-center motion is set to be constant relative to the base in  $x_b$  direction, however, it can also be user-defined to achieve more complex applications. Furthermore, for the centroidal motion height control, even we use the centroidal motion relative to the base as the reference, the whole-body motion generator still generates the desired centroidal translational motion in the vertical direction of the inertial frame. Therefore, the desired spatial centroidal acceleration serves as the input of the feed-forward torque controller in Section 4.4.2 and the centroidal spatial motion Jacobian  $\mathbf{J}_G$  can be applied directly. In the feedback torque controller in Section 4.4.1, we control the centroidal height motion relative to the base directly. In addition, the proposed altitude control model can also be applied to conventional quadruped robots without leg-massless assumption.

## 4.4 Whole-body Torque Controller

In this section, several impedance controllers are defined to enable the robot with compliance abilities in which the desired positions and velocities are applied. These feedback controllers are integrated into the prioritized impedance controller based hierarchical torque control in Chapter 3. We have developed the wheel motion generator in Section 4.2 and the whole-body generator in Section 4.3, for the rolling based locomotion on rough terrain, we control two tasks which are ranked in the priority order from high to low as: 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. It is noticed that the wheel motion is adaptive with the legged-joint motion. In addition, the contact constraints account for 12 DOFs. Therefore, the whole-body DOF  $n = 22$  is accounted. The desired accelerations of the  $G$  and  $w$  tasks serve as the inputs of the

feed-forward torque controller. For the wheel motion generator in Section 4.2, we apply the impedance controllers defined in Section 3.3, except the one relating to the manipulator in (3.6). Therefore, in this section, we focus on the establishment of impedance controllers relating to the whole-body motion generator in Section 4.3.

#### 4.4.1 Feedback Whole-body Impedance Controllers

When the robot runs in 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, each impedance controller  $\boldsymbol{\tau}_{im_k} \in \mathbb{R}^n$  is projected into the generalized coordinate space. In this section, all the superscript  $d$  means the desired values of any vector and these desired vectors are generated in the whole-body motion generators or they serve as the motion references. Initially, we define one impedance controller for the whole-body angular and translational momentums to track the robot centroidal motion except the altitude control. Since the angular position of the whole-body CoM has no sense, we apply only the damping matrix  $\mathbf{D}_\omega \in \mathbb{R}^{3 \times 3}$  for the angular momentum tracking as follows,

$$\boldsymbol{\tau}_{im_G} = (\mathbf{S}_u \mathbf{J}_G)^T \left[ \mathbf{K}_v \begin{pmatrix} \mathbf{p}_{G[x]}^d - \mathbf{p}_{G[x]} \\ \mathbf{p}_{G[y]}^d - \mathbf{p}_{G[y]} \end{pmatrix} + \mathbf{D}_v \begin{pmatrix} \mathbf{v}_{G[x]}^d - \mathbf{v}_{G[x]} \\ \mathbf{v}_{G[y]}^d - \mathbf{v}_{G[y]} \end{pmatrix} \right], \quad (4.30)$$

where  $\mathbf{S}_u = [\mathbf{1}_{5 \times 5} \ \mathbf{0}_{5 \times 1}]$  is used to select the first 5 rows of  $\mathbf{J}_G$ , because in the inertial frame, only the centroidal translational motion in the forward and lateral directions is controlled.  $\mathbf{K}_v \in \mathbb{R}^{2 \times 2}$  and  $\mathbf{D}_v \in \mathbb{R}^{2 \times 2}$  denote the stiffness and damping matrices for the robot centroidal translational motion. Another impedance controller for the centroidal height control relative to the base frame should be defined as below,

$$\boldsymbol{\tau}_{im_{\tilde{c}}} = \mathbf{J}_{\tilde{c}}^T \left[ K_{\tilde{c}} \left( \tilde{p}_{G[z]}^d - \tilde{p}_{G[z]} \right) + D_{\tilde{c}} \left( \dot{\tilde{p}}_{G[z]}^d - \dot{\tilde{p}}_{G[z]} \right) \right], \quad (4.31)$$

where  $\mathbf{J}_{\tilde{c}} = [\mathbf{0}_{1 \times 6} \ \mathbf{S}_d \mathbf{J}_G^g]$  is the Jacobian for the centroidal translational motion relative to the base frame in  $z_b$  direction.  $\mathbf{S}_d = [\mathbf{0}_{1 \times 5} \ 1]$  is used to select the last row of  $\mathbf{J}_G^g$  in Section 5.42.  $K_{\tilde{c}}$  and  $D_{\tilde{c}}$  denote the relative stiffness and damping scalars.

In addition, one impedance controller is designed for the wheel motion as follows,

$$\boldsymbol{\tau}_{im_w} = \mathbf{J}_w^T \left[ \mathbf{K}_w (\mathbf{q}_w^d - \mathbf{q}_w) + \mathbf{D}_w (\dot{\mathbf{q}}_w^d - \dot{\mathbf{q}}_w) \right], \quad (4.32)$$

where  $\mathbf{J}_w \in \mathbb{R}^{n_c \times n}$  represents the Jacobian relating to all wheel-joint motion.  $\mathbf{K}_w \in \mathbb{R}^{n_c \times n_c}$  and  $\mathbf{D}_w \in \mathbb{R}^{n_c \times n_c}$  denote the stiffness and damping for the wheel motion respectively.

Another impedance controller is defined to enable the base with small roll angular as follows,

$$\boldsymbol{\tau}_{im_b} = \mathbf{J}_{b_x}^T \left[ K_b \left( \boldsymbol{\theta}_{b[x]}^d - \boldsymbol{\theta}_{b[x]} \right) + D_b \left( \boldsymbol{\omega}_{b[x]}^d - \boldsymbol{\omega}_{b[x]} \right) \right], \quad (4.33)$$

where  $\mathbf{J}_{b_x} \in \mathbb{R}^{1 \times n}$  is the Jacobian for the base angular motion around  $x_b$  axis. The desired roll angle  $\boldsymbol{\theta}_{b[x]}^d$  is set to be zero.  $K_b$  and  $D_b$  denote respectively the relative angular stiffness and damping scalars.

Furthermore, the compliance for the legged suspension plays a significant role for the robot to cross over various rough terrains. Therefore, we establish one impedance controller on all leg-end (wheel-center) points as below,

$$\boldsymbol{\tau}_{im_x} = \mathbf{J}_x^T \left[ K_x \left( \tilde{\mathbf{p}}_{w[x]}^d - \tilde{\mathbf{p}}_{w[x]} \right) + D_x \left( \dot{\tilde{\mathbf{p}}}_{w[x]}^d - \dot{\tilde{\mathbf{p}}}_{w[x]} \right) \right], \quad (4.34a)$$

$$\boldsymbol{\tau}_{im_z} = \mathbf{J}_z^T \left[ -\mathbf{K}_z \begin{pmatrix} \tilde{\mathbf{p}}_{w_1[z]} - \tilde{\mathbf{p}}_{w_2[z]} \\ \tilde{\mathbf{p}}_{w_1[z]} - \tilde{\mathbf{p}}_{w_3[z]} \\ \tilde{\mathbf{p}}_{w_1[z]} - \tilde{\mathbf{p}}_{w_4[z]} \end{pmatrix} - \mathbf{D}_z \begin{pmatrix} \dot{\tilde{\mathbf{p}}}_{w_1[z]} - \dot{\tilde{\mathbf{p}}}_{w_2[z]} \\ \dot{\tilde{\mathbf{p}}}_{w_1[z]} - \dot{\tilde{\mathbf{p}}}_{w_3[z]} \\ \dot{\tilde{\mathbf{p}}}_{w_1[z]} - \dot{\tilde{\mathbf{p}}}_{w_4[z]} \end{pmatrix} \right], \quad (4.34b)$$



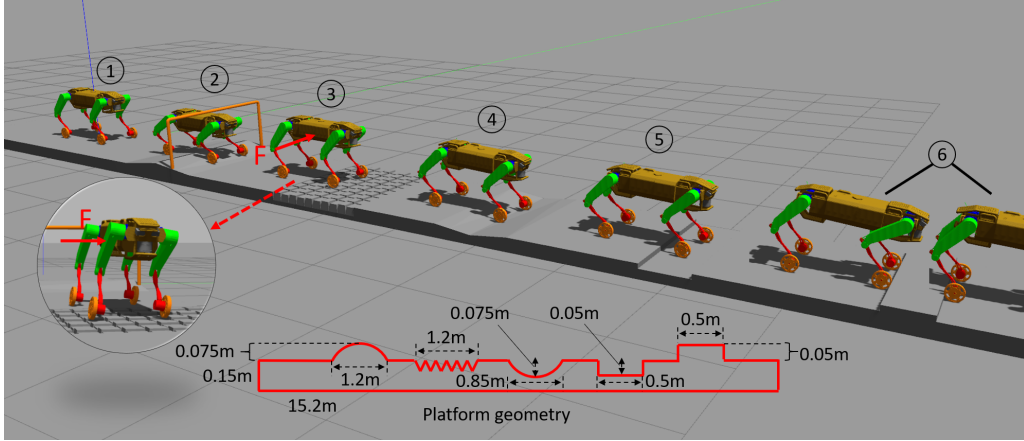


Figure 4.4: The robot runs forward on various rough terrains and handles several activities. The platform *height* = 0.15m, *length* = 15.2m, and the rough terrain geometry is depicted. The whole running process is divided into six stages.

where  $\tilde{\mathbf{p}}_{w[x]} = [\tilde{\mathbf{p}}_{w_1[x]} \ \tilde{\mathbf{p}}_{w_2[x]} \ \tilde{\mathbf{p}}_{w_3[x]} \ \tilde{\mathbf{p}}_{w_4[x]}]^T$  in (4.34a) combines all the wheel-center positions relative to the base frame in  $x_b$  direction, and  $\mathbf{J}_x \in \mathbb{R}^{n_c \times n}$ ,  $K_x$  and  $D_x$  denote the relative Jacobian, the stiffness and damping scalars. The impedance controller in (4.34b) is used to enable the four wheel-center positions to be similar in the vertical direction of the base frame, and  $\mathbf{J}_z \in \mathbb{R}^{3 \times n}$ ,  $\mathbf{K}_z \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{D}_z \in \mathbb{R}^{3 \times 3}$  denote the relative Jacobian, the stiffness and damping matrices.  $\mathbf{J}_x$  and  $\mathbf{J}_z$  are achieved as follows,

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{0}_{1 \times 6} & \mathbf{J}_{x_1} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 6} & \mathbf{0}_{1 \times 4} & \mathbf{J}_{x_2} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 6} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{J}_{x_3} & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 6} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{J}_{x_4} \end{bmatrix}, \quad (4.35a)$$

$$\mathbf{J}_z = \begin{bmatrix} \mathbf{0}_{1 \times 6} & \mathbf{J}_{z_1} & -\mathbf{J}_{z_2} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 6} & \mathbf{J}_{z_1} & \mathbf{0}_{1 \times 4} & -\mathbf{J}_{z_3} & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 6} & \mathbf{J}_{z_1} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & -\mathbf{J}_{z_4} \end{bmatrix}, \quad (4.35b)$$

where  $\mathbf{J}_{x_i} = [\mathbf{S}_x \tilde{\mathbf{J}}_{w_i} \ \mathbf{0}] \in \mathbb{R}^{1 \times 4}$  and  $\mathbf{J}_{z_i} = [\mathbf{S}_z \tilde{\mathbf{J}}_{w_i} \ \mathbf{0}] \in \mathbb{R}^{1 \times 4}$ .  $\tilde{\mathbf{J}}_{w_i}$  is the Jacobian relating to the wheel  $i$  center motion relative to the base in Section 4.3.2.

#### 4.4.2 PIC based Torque Controller

The prioritized impedance controller in Chapter 3 is detailed for the application in this chapter as follows,

$$\mathbf{\Gamma} = \left[ \sum_{t=1}^K \mathbf{J}_{t|pre(t)}^T \bar{\mathbf{J}}_{t|pre(t)}^T \right] (\boldsymbol{\tau}_{im_G} + \boldsymbol{\tau}_{im_{\bar{G}}} + \boldsymbol{\tau}_{im_b} + \boldsymbol{\tau}_{im_w} + \boldsymbol{\tau}_{im_x} + \boldsymbol{\tau}_{im_z}), \quad (4.36)$$

where  $K = 2$ ,  $\mathbf{J}_1 = \mathbf{J}_G$  and  $\mathbf{J}_2 = \mathbf{J}_w$  for our application. In this way, various impedance force are integrated into only one item  $\mathbf{\Gamma}$  which influences the defined two tasks in a decoupled way. Furthermore,  $\mathbf{\Gamma}$  can integrate the impedance controllers to the designed tasks, e.g. including the centroidal motion task in (4.30)(4.31) and the wheel motion task in (4.32). Especially, several general impedance controllers indirectly acting on the designed tasks, e.g. the impedance controllers in (4.33), (4.34a) and (4.34b), can also be integrated into the single torque in (4.36). The PIC based pseudo-inverse based inverse dynamics method in Section 3.4.1 is applied in the first two simulations in Section 4.5.1 and Section 4.5.2. The P-HQP torque controller in Section 3.4.3 is applied in the simulation scenarios in Section 4.5.3 and Section 4.5.4. The two torque controllers integrate the feedback and feed-forward torque controllers together.



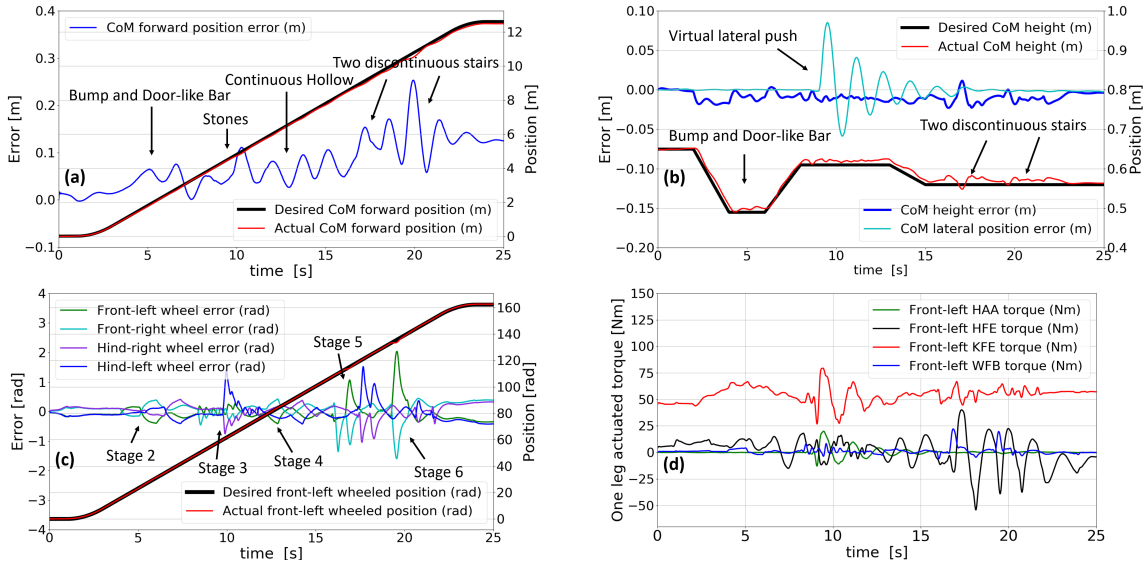


Figure 4.5: Motion tracking results of the centroidal motion and the wheel motion. The left and right axes represent the errors and positions respectively. (a) records the robot centroidal forward position and error. (b) plots the lateral position error and its vertical height/error. (c) shows the four-wheel 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.

## 4.5 Simulation on Various Rough Terrains

The wheel motion generator in Section 4.2 and the whole-body motion generator in Section 4.3 are tested in several simulation scenarios using our virtual quadruped-on-wheel robot, “*TowrISIR*” in Fig. 2.1. The simulation results are also reported using the physical engine established by ROS-GAZEBO and the similar sensors applied in Chapter 3. Based on the sensor-feedback data, we calculate the properties relating to the base and the centroidal motions. These robot states are applied in the impedance controllers and the motion generators, shown in Fig. 4.1. We set the centroidal motion reference in the sagittal plane and its lateral reference is always zero. The motion simulations of the wheel-motion generator and the whole-body motion generator are based on the general control frameworks presented in Fig. 4.1 and Fig. 4.2 respectively, which both combine the feedback controller in the impedance input-port and the feed-forward controller that computes the required torque which satisfies the Equation of Motion, the contact constraints and the task hierarchy. The first two simulation scenarios in this chapter can be seen in the video<sup>1</sup>. The second two simulation scenarios are referred in another video<sup>2</sup>.

### 4.5.1 Running on Artificial Environment

In the first scenario in Fig. 4.4, 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. [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. 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

<sup>1</sup>See the first and second simulation videos in Chapter 4: [▶◊◊◊◊◊◄](#)

<sup>2</sup>See the third and fourth simulation videos in Chapter 4: [▶◊◊◊◊◊◄](#)

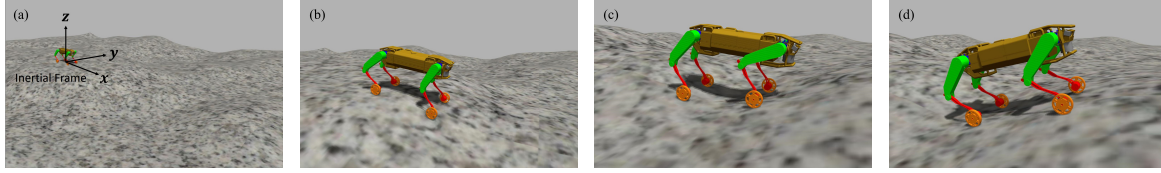


Figure 4.6: The robot is driven on unknown 3D rough terrain with various slopes. (a) is the beginning state. In (b), the robot runs downward one slope. In (c) and (d), the robot prepares and climbs upward respectively. (b) and (c) show the legged suspension with compliance.

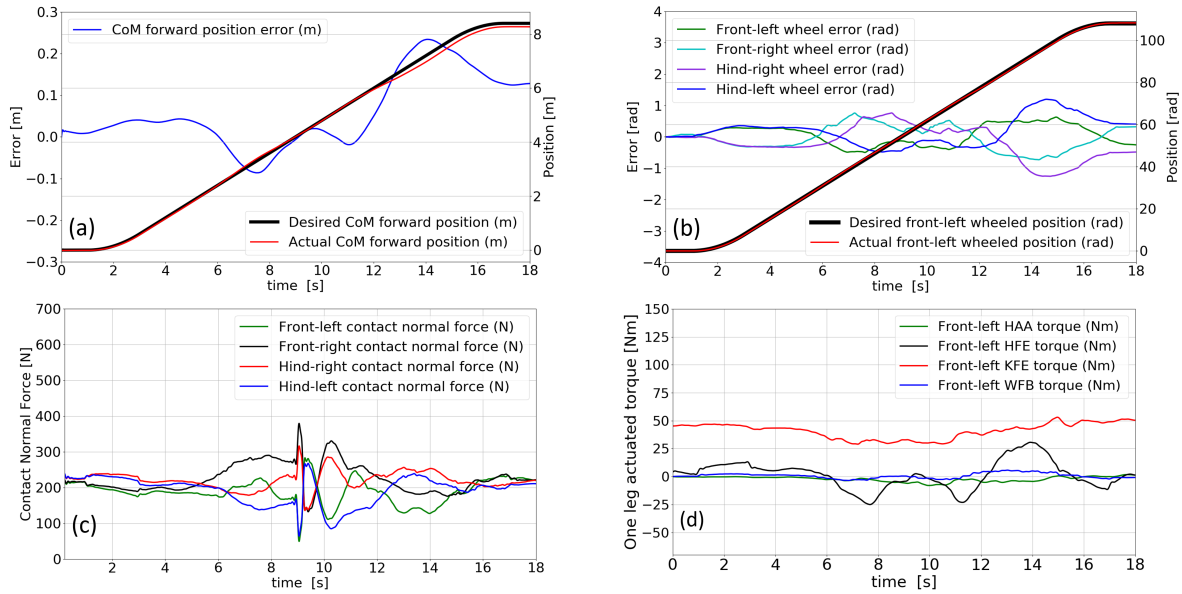


Figure 4.7: Simulation results of the second scenario. (a) records the quadruped CoM motion in forward direction. (b) plots the four wheels' rotational position errors along the whole process. Only the desired and actual front-left wheel positions are plotted. (c) shows the four wheel contact normal forces. (d) plots the actuated-joint torque for the front-left leg.

to squat down and rolls over the bump and through the bar. As described in Section 3.3, 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 2.1.3, we can see that the wheel tracking in Fig. 4.5(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 2.1.3 and the generalized dynamics model. Then the imprecise models influence the outputs of the motion generator in Section 4.2 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 almost coincident in the long running process, shown in Fig. 4.5(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. 4.5(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 4.2. By using the PIC

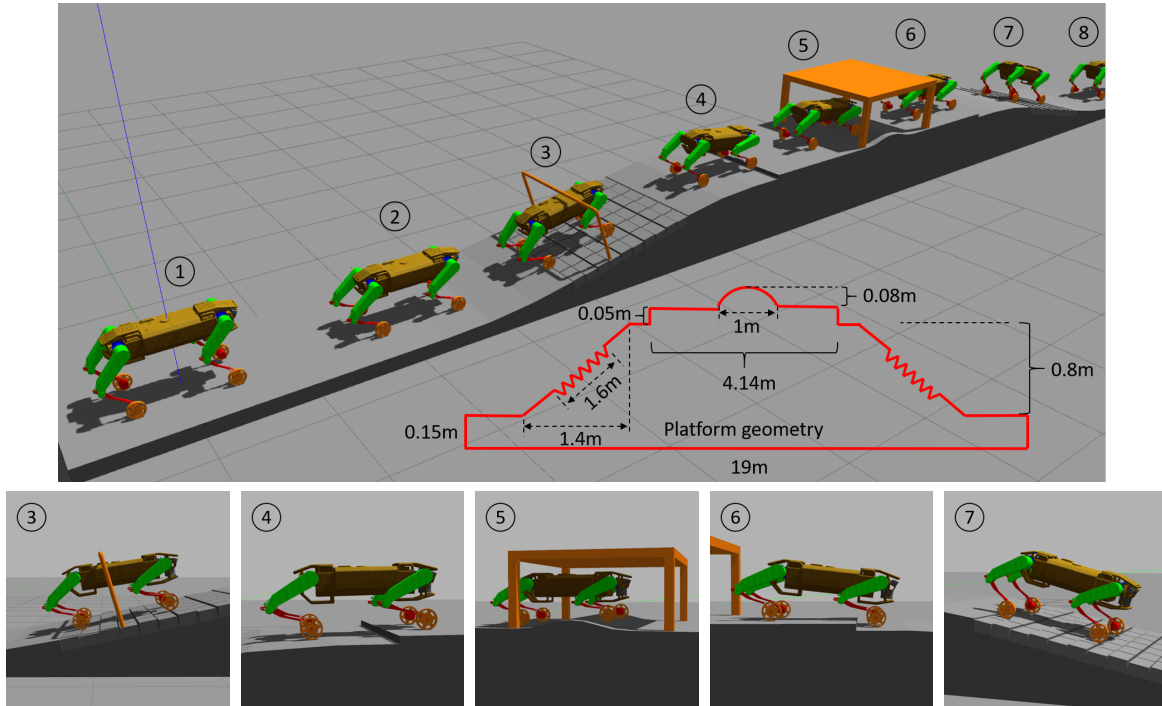


Figure 4.8: The robot runs forward on various rough terrains and handles several activities. The symmetric rough terrain geometry of the platform is depicted on the picture. The whole running process is divided into eight stages. We also show some zoom-in pictures of several stages.

based whole-body torque control in Section 3.4.1, 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. 4.5(b). In addition, we plot the actuated torque for the joints of the front-left leg in Fig. 4.5(d). 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.

#### 4.5.2 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. 4.6(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. 4.6(b) or prepares the climb-upward motion in Fig. 4.6(c). Finally, the robot climbs over the slope in Fig. 4.6(d). The CoM and wheel positions and errors are plotted in Fig. 4.7(a) and Fig. 4.7(b) respectively. Even with different legged suspension and configuration, our wheel-motion generator in Section 4.2 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 pseudo-inverse based inverse dynamics controller in Section 3.4.1, the required actuated torque is generated to track the motion references. The contact constraint with the highest priority is ensured, shown in Fig. 4.7(c) which shows that the four legs are always in contact with the ground, and the four normal contact forces are around 200N. In Fig. 4.7(d), we plot the actuated torque for the joints of the

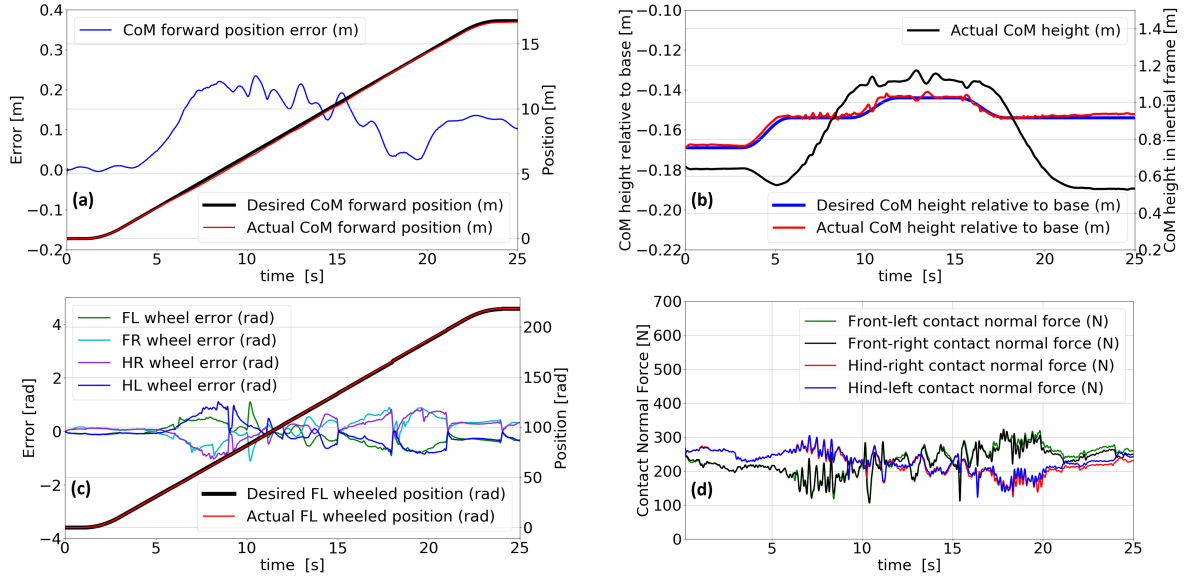


Figure 4.9: Motion tracking results of the centroidal motion. The left and right axes of (a) and (c) represent the errors and positions respectively. (a) records the robot centroidal forward position and error. (b) plots the desired and actual CoM position relative to the base in the base vertical direction. (b) also plots the actual CoM height position in the inertial frame. (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 four wheel contact normal forces.

front-left leg, which is generated by our compliant torque controller in Section 3.4.1. It is noticed that compared with the generated joint torque in Section 4.5.3, 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.

### 4.5.3 Running on Terrain with Big Altitude Difference

In the simulation scenario in Fig. 4.8, only the positions of the door-like horizontal bar and the desk are known. The robot crosses over several kinds of artificial terrains, including two slopes with stones, two stairs, one bump. To validate the effectiveness and adaptivity of our whole-body motion generator and the altitude control model, the robot is driven on the rough terrain with a much-difference altitude. The platform geometry is shown in Fig. 4.8 and the max altitude difference is 0.8m which is higher than the robot normal height. We show the whole process with eight stages in Fig. 4.8, and the zoom-in pictures in the middle-process behaviors are shown in Fig. 4.8. [Stage 1]: the beginning state. [Stage 2]: the robot squats down to prepare to cross through one door-like horizontal bar. [Stage 3]: the robot is driven upward one slope with stones (the max difference height of the stones is 0.01m). [Stage 4]: the robot crosses upward one stair. [Stage 5]: the robot squats down more to cross through one desk and over one bump under the desk. [Stage 6]: The robot is driven downward one stair. [Stage 7]: The robot runs downward one slope with stones. [Stage 8]: The robot stops on the flat platform. Since the platform geometry is unknown and the contact points are always estimated at the lowest points of the wheels in Section 2.1.3, we can see that the wheel tracking in Fig. 4.9(c) experiences a relatively big fluctuation. It is not very precise when the robot runs on slopes with stones in stage 3 and 7, discontinuous stairs in stage 4 and 6, and the continuous bump in stage 5. Even the imprecise models influence the outputs of the motion generator in Section 4.3 and the results of the centroidal forward motion tracking, the desired and actual CoM positions in the forward direction are still almost coincident in the long running process, shown in Fig. 4.9(a). During the process in stage 2, 3 and 5, we control the CoM motion relative to the base to enable the robot to squat down to cross the door-like horizontal bar and the desk (see Fig.

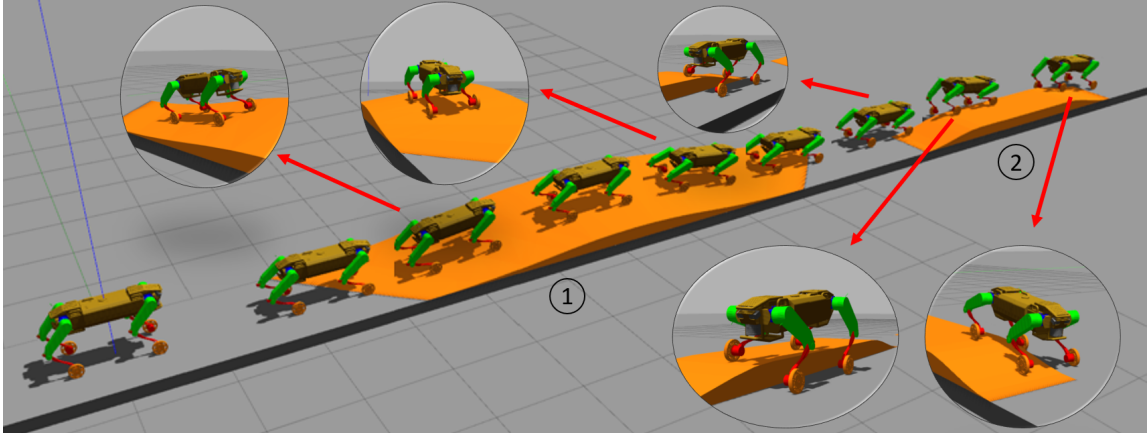


Figure 4.10: The robot runs forward on the unsymmetrical terrain, including (1) the slope in both forward and lateral directions, and (2) one platform with the half width of the road. The upper zoom-in pictures show the robot behavior on the slope indexed with (1). The left edge of the slope is  $0.3m$  higher than the right edge. The bottom zoom-in pictures show that only the right two legs are the platform with index (2). The height of the platform (2) is  $0.2m$ .

4.9(b)). The CoM motion relative to the base also enables the robot to run on slopes and on different altitude geometry without knowing the environment, and we can see the tracking performance of this relative position  $\tilde{\boldsymbol{p}}_{G[z]}$  is quite good with very small errors. We also plot the robot CoM height in the inertial frame, and its profile is almost consistent with the platform geometry. The adaptive whole-body motion is generated to track the CoM motion according to the real-time robot states, due to the fact that the actual robot configuration is used to achieve the whole-body velocity reference, and the actual robot configuration/velocity is applied to derive the whole-body acceleration reference, which enables the robot to adapt to the rough terrain with robustness, detailed in Section 4.3. By using the P-HQP controller in Section 3.4.3 integrated with our various impedance controllers in Section 4.4.1, the robot is enabled to run on various rough terrains and track the centroidal motion with compliance, especially for the periods with stones in stage 3 and 7, the stairs in stage 4 and 6, and the bump in stage 5, and the compliance behaviors are shown in Fig. 4.8. In addition, the contact constraints are ensured with the higher priority than the designed two tasks, and Fig. 4.9(d) shows that the four legs are always in contact with the ground and the normal contact forces are estimated using (3.31) and (3.32).

#### 4.5.4 Running on Non-symmetric Geometry

In this simulation scenario, shown in Fig. 4.10, the robot evolves on the unknown unsymmetrical geometry, including one slope in the forward and lateral directions, and one platform that is put on the right half of the terrain. The altitude difference of the geometry is less than that in Fig. 4.8, but has unsymmetrical layout, which enforces the robot to perform the adaptive behaviors. We show the whole running process and the zoom-in middle-process pictures, shown in Fig. 4.10. The tracking performance of the CoM position in the forward direction is quite satisfactory, shown in Fig. 4.11(a). For the centroidal position relative to the base in  $z_b$  direction, we set the reference to be the same with the simulation in Fig. 4.8, and the tracking performance is better by comparing the results in Fig. 4.9(b) and Fig. 4.11(b), due to the fact that the terrain in this simulation is continuous in the whole process. Since the geometry is much smoother than that in Fig. 4.8, we can see that the wheel tracking in Fig. 4.11(c) experiences a relatively smaller fluctuation. We use this simulation to further verify the robust whole-body motion generator and the compliant torque controller P-HQP in Section 3.4.3. The proposed motion generator prompts the whole-body adaptive behaviors depending on the actual robot states, detailed in Section 4.3, and it enables the robot to run on the unknown complex slopes (1), and handle the compatible behaviors on the platform (2) with only two legs touching on it, shown in the zoom-in pictures in Fig.



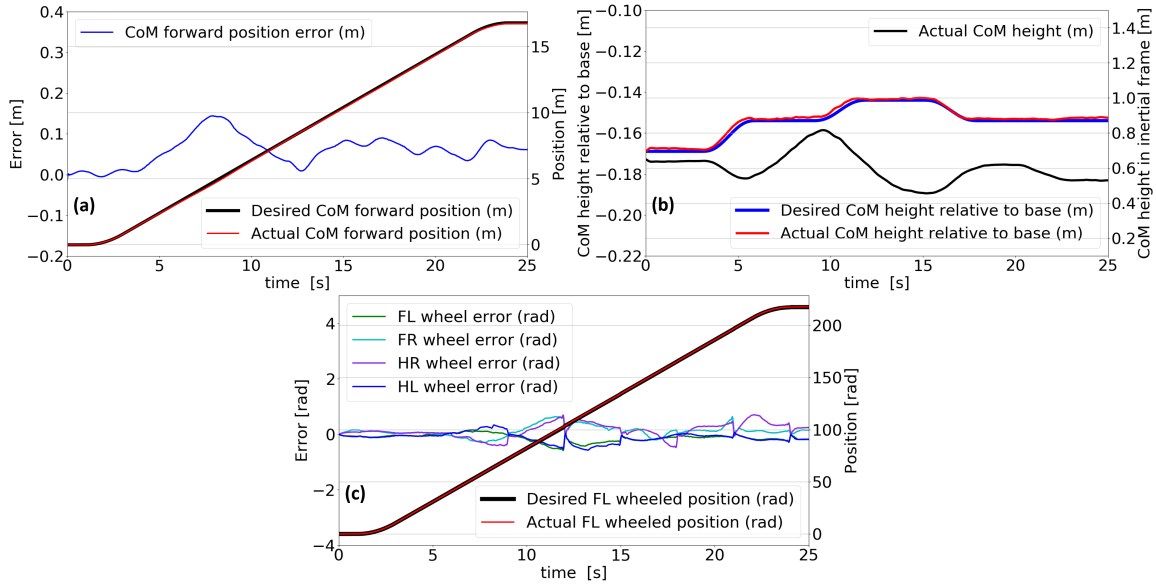


Figure 4.11: Two-task tracking performance relating to the centroidal motion and the wheel motion. The three figures have the same descriptions as in the caption of Fig. 4.9.

4.10. In addition, along with the motion tracking of the two tasks, the wheel-contact constraints are always ensured by the proposed PIC and our optimization based prioritized torque controller.

To sum up, the simulations verify that the proposed wheel-motion generator, the whole-body motion generator and the altitude control model are adaptive and effective for our robot when running on various unknown rough terrains, even with a much-altitude difference. The PIC based pseudo-inverse dynamics controller and the PIC based multi-task operational space controller P-HQP enable the robot with compliance abilities, especially for active legged suspension. The simulations are also done on a usual PC with an *i7-7700* CPU, *3.6GHz*. The sampling time is fixed with *2ms*. For the optimization base torque controller, one QP is solved at each time step which has 16 decision variables and 68 inequality constraints. The maximum calculation period is *0.9ms*, which is efficient and suitable for real-time control of other locomotion robots with high DOFs. The simulation results provided by our approach are quite satisfactory, and they can also be seen in the video that shows the real-time tracking performance.

## 4.6 Summary

In this chapter, we propose two motion generators for a quadruped-on-wheel robot, *TowrISIR*, including one wheel-motion generator and one new whole-body motion generator which both track the centroidal motion using the robot rolling-mode locomotion. The whole-body inverse kinematics model is decomposed into components relating to the base motion, the legged motion and the wheel motion. In wheel motion generator, we assume that the desired legged suspension is already achieved, and the wheel motion is derived by combining the centroidal momentum & dynamics model and the whole-body kinematics model. In the second whole-body motion generator, the desired legged suspension can be generated online. The centroidal motion relative to the base frame along  $z_b$  direction is controlled using one proposed altitude control model to enable the robot to run on terrains with a much-difference altitude. Then the whole-body motion generator is established by combining the whole-body inverse kinematics model, the wheel-center motion model, the altitude control model and the centroidal momentum/dynamics model. By using two prioritized impedance controller based torque controllers in Chapter 3, the contact constraints are ensured and our robot is successfully driven to track the centroidal motion reference and behave compatibly on various simulated rough terrains, which validate the adaptivity and effectiveness of our two motion generators.

# Model-based Legged Locomotion

---

## Contents

<b>5.1</b>	<b>Introduction and Chapter Outline</b>	<b>75</b>
<b>5.2</b>	<b>Dynamics Modeling</b>	<b>76</b>
5.2.1	Whole-Robot Dynamics Modeling	76
5.2.2	Each Leg Centroidal Motion	81
5.2.3	Whole-body Centroidal Jacobian	83
5.2.4	Centroidal Model Decompositions	84
<b>5.3</b>	<b>Legged Motion Generation</b>	<b>85</b>
5.3.1	New Quadruped Locomotion Principle	85
5.3.2	New Legged Motion Generator	86
5.3.3	External Motion Reference	87
5.3.4	Internal Motion Reference	89
5.3.5	State Machine	92
5.3.6	Modified Centroidal Models For Swing-leg Motion Generators	93
5.3.7	Desired Composite Swing-leg Motion	94
5.3.8	Desired Swing-leg Motion for Walking	95
5.3.9	Leg-Motion Relationship for Trotting	96
5.3.10	Desired Swing-leg Motions for Trotting	97
<b>5.4</b>	<b>Whole-body Torque Controller</b>	<b>98</b>
5.4.1	Prioritized Compliant Controller	99
5.4.2	Optimization based Torque Controller	101
5.4.3	Lateral Push-recovery Reaction	103
<b>5.5</b>	<b>Simulations for QuadISIR</b>	<b>104</b>
5.5.1	Trotting Forward	105
5.5.2	Trotting Lateral, Turning in Place and Push-recovery	108
5.5.3	Trotting on Rough Terrain	109
<b>5.6</b>	<b>Summary</b>	<b>111</b>

---

This chapter proposes a novel legged locomotion principle for tracking the quadruped-robot base spatial motion which can generate the trotting and walking gaits automatically. The principle is based on a new legged locomotion generator using the robot centroidal properties without a massless-leg assumption. Our quadruped robot is newly designed with symmetric legged dynamics properties using parallelogram mechanisms, and the corresponding generalized dynamics model is built. The centroidal momentum and dynamics models are established by combining the centroidal properties of the base and each leg, in which the virtual joint between the base and each leg CoM is used instead of the real joints. The stance legs follow the base motion with respect to contact constraints. The swing legs compensate for the delay influences (to centroidal motion) by the stance legs. By combining the whole-body kinematics model and the kinematics constraints relating to the contact constraints and the parallelogram mechanism, the adaptive swing-leg motion is achieved in the running process without pre-defined trajectories. When two swing legs land at different timings, the walking gait is applied. We also propose a new state machine for switching the locomotion gaits between walking and trotting, which also serves the torque

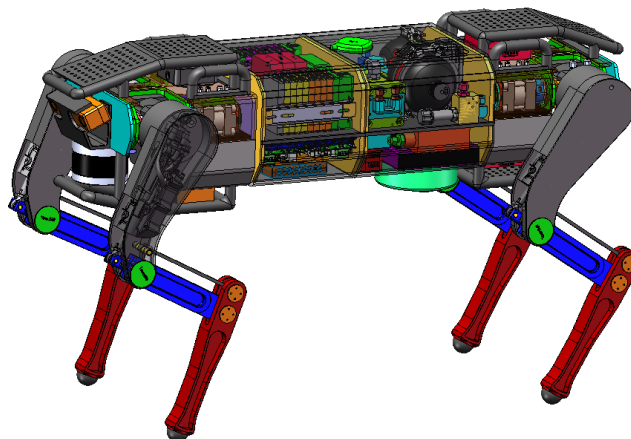


Figure 5.1: “QuadISIR”—A quadruped robot developed in ISIR lab. Each leg has three active-actuated joints, including the hip abduction/adduction joint (HAA), hip flexion/extension joint (HFE), knee flexion/extension joint (KFE). The fourth ankle joint is actuated by one mechanism structure.

controller which determines the dimension of the legged control states and the contact Jacobian. We apply our hierarchical torque controller P-HQP to enable the robot with compliant dynamics properties. Therefore, our new locomotion principle integrates the system design, motion generation and whole-body torque control together, so that this one common framework is able to handle several locomotion gaits. To verify the usefulness and validity of our new locomotion principle based legged motion generator, we run several simulations: walking, trotting, turning, recovering from big lateral push force on the base.

## 5.1 Introduction and Chapter Outline

To give complex legged robots the ability of crossing rough terrain with highly dynamic behaviors, the motion generators play a big role in addition to the low level dynamics controller. The complexity of legged systems is a key problem on how to achieve dynamically-stable locomotion in changing environments. During the last decades, several model-based methods have been developed to generate the legged motion, as detailed in the state of the art relating to the legged motion generation in Chapter Introduction. In the quadruped robotics field, the legs are always assumed to be massless which is unreasonable. Even if the legs only account for 20% of the whole-body weight, they experience much bigger accelerations than the floating base. In fact, the inertia forces for the base and one leg are on the similar level. Furthermore, for efficient locomotion, with the massless-leg assumption, the joint torque cannot be optimized since infinite acceleration can be derived using Newton’s second law. In addition, the leg trajectories are always defined initially which is far from natural locomotion principle and it is not realistic to efficiently handle the complex changing environment.

In this chapter, we propose a new locomotion principle for quadruped robots without the massless-leg assumption, which further closes the gap between the quadruped robots with their counterparts. For the walking and trotting locomotion gaits, the whole-body CoM or the base motion is defined initially, which is consistent with the locomotion logic. Then the stance legs follow the base motion. The swing legs compensate the delayed influence from the stance-leg motion to the centroidal motion. Based on this principle, the centroidal momentum/dynamics models are established using the base and each-leg centroidal motion. The centroidal translational models are used for the walking mode, and the spatial centroidal models are applied for the trotting mode. The swing-leg motion is achieved in the running process without requiring pre-defined trajectories. The principle can also be applied to enable the robot to handle turning and push-recovery motions. Finally, we use our developed hybrid hierarchical and prioritized weighted optimization method to achieve the whole-body torque control and enable the robot



to run on various rough terrains.

The advantages of the new locomotion principle: It is more logical to define the base or the centroidal motion reference, and then the generated legged motions follow the base/CoM reference. That is a desirable property that is shared with vehicles for which the wheeled motion are not necessary to be defined. The legged motion of quadruped robots should also be generated on-line. The benefits of getting rid of the leg-massless assumption: The legs can be designed to have bigger mass to be more powerful which is important for more serious jobs, e.g. rescue and carrying works. The distance between the robot CoM and its base in the vertical direction can be bigger which is more convenient to adjust the impedance parameters relating to its feedback control. The strengths without pre-defined leg trajectories: For normal terrains, the robot can achieve the common trotting actions in forward, lateral, rotational turning and push recovery without requiring to define the legged trajectories, even on relatively rough terrains. The step length is automatically generated which is suitable for starting to trot and decelerating to stop. Compared with the state of the art, the proposed state machine determines the switch of the swing-leg indexes, the dimensions of the legged control states and the contact Jacobian in the torque controller. Our new locomotion principle integrates the system design, the motion generation and the torque control together which synthesizes trotting, walking, turning and push recovery using only one control framework.

The remainder of this chapter is organized as follows: In Section 5.2, we establish the system modeling and decompose the centroidal momentum/dynamics model in a new way. In Section 5.3, a novel legged locomotion principle is proposed by a novel legged locomotion generator without the specific legged trajectory definition. In Section 5.4, we build our compliant torque controller based on P-HQP in Chapter 3, in which the kinematic constraints relating to the parallelogram mechanism is integrated. In Section 5.5, we validate and discuss our method through numerical simulations. Section 5.6 closes the chapter with conclusions.

## 5.2 Dynamics Modeling

Quad*ISIR* is a new quadruped robot that is designed in ISIR laboratory, shown in Fig. 5.1. It is equipped with four legs, each leg has four joints, including HAA (hip abduction & adduction) joint, HFE (hip flexion & extension) joint, KFE (knee flexion & extension) joint and one Ankle joint, shown in Fig 5.2. Three of these joints are actively actuated. The fourth ankle joint is actuated using one parallelogram mechanism resulting in the KFE and ankle joint angles being of opposite values. The robot mass is about  $100kg$  with length  $\approx 1.06m$ , width  $\approx 0.55m$  and normal height  $\approx 0.8m$ .

In this section, the robot generalized dynamics model is first introduced for our robot with the new-designed leg. Then each-leg centroidal properties are achieved. Finally the robot centroidal momentum and dynamics models are decomposed by using each leg-CoM motion instead of real leg bodies. Spatial vectors [Featherstone 2014] are used throughout this chapter written in upright bold letters, and conventional 3D vectors are written with italic ones. The cross product function for spatial vectors and conventional 3D vectors are also referred in [Featherstone 2014].

### 5.2.1 Whole-Robot Dynamics Modeling

#### 5.2.1.1 Generalized Dynamics Model:

Although each leg has only 3 actuated joints, we build the generalized dynamics model by considering the four joints of each leg. For the parallelogram mechanism, since the parallel-rod mass is very small, we treat this robot only with tree configuration and neglect the parallel rod. This will be treated as model inaccuracy and compensated by feedback based controllers. Therefore, we can derive the inertia matrix, centripetal & coriolis force and gravity force using Composite Rigid Body algorithm [Walker 1982] and Recursive Newton-Euler algorithm [Luh 1980] respectively. For the whole robot considering four legs, we apply this approach to achieve the generalized dynamics model, which will be used to derive the prioritized multi-task operational space torque controller. From the robot structure in Fig. 5.1 and the

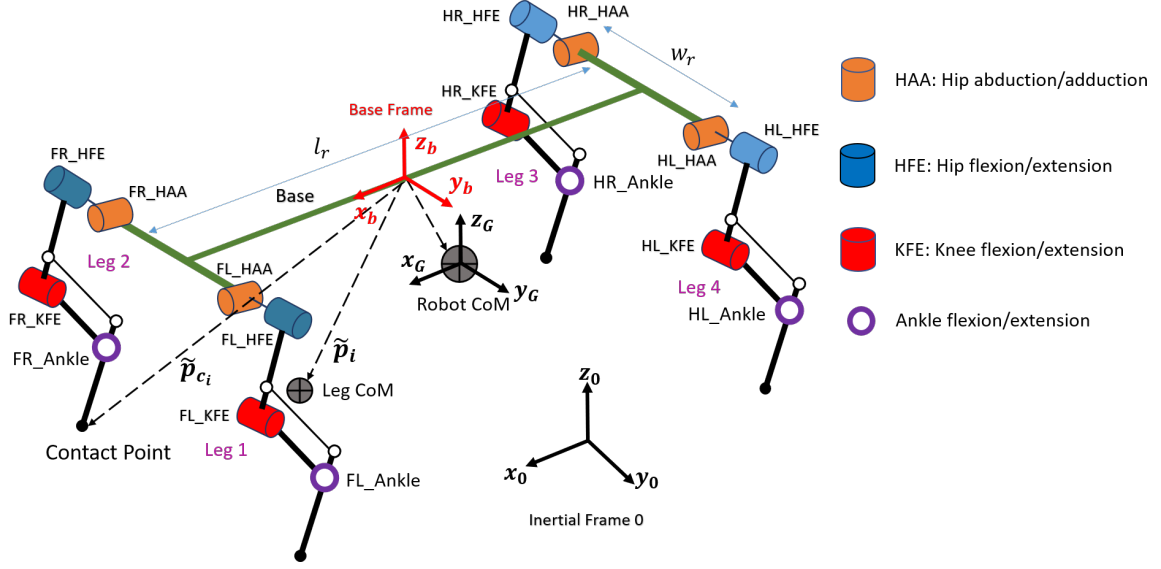


Figure 5.2: Robot Configuration: Each leg is consisted with three actuated joints, HAA, HFE and KFE, whose frame  $z$  axes are consistent with the joint axes. F, L, R, H represent front, left, right, hind.

configuration in Fig. 5.2, the robot configuration is defined as follows,

$$\mathbf{q} = [\mathbf{x}_b^T \quad \mathbf{q}_{r_1}^T \quad \mathbf{q}_{r_2}^T \quad \mathbf{q}_{r_3}^T \quad \mathbf{q}_{r_4}^T]^T \in \mathbb{R}^{n_b+4n_l=n}, \quad (5.1)$$

where  $\mathbf{x}_b = [\boldsymbol{\theta}_b^T \quad \mathbf{p}_b^T]^T$  is the orientation and position of the floating base in inertial global coordinate so that  $n_b = 6$ . The robot has  $n_l = 4$  legs, so that  $n = 4n_l + n_b = 22$  denotes the robot DOF number.  $\mathbf{q}_{r_i} \in \mathbb{R}^4$  denotes the four-joints configuration of leg  $i$  as follows,

$$\mathbf{q}_{r_i} = [\mathbf{q}_{HAA_i} \quad \mathbf{q}_{HFE_i} \quad \mathbf{q}_{KFE_i} \quad \mathbf{q}_{Ankle_i}]^T, \quad (5.2)$$

where the leg  $i$  KFE joint and Ankle joint has the opposite values,  $\mathbf{q}_{KFE_i} = -\mathbf{q}_{Ankle_i}$ . In addition, we define  $\mathbf{q}_{a_i}$  to include the leg  $i$  actuated-joint configuration as follows,

$$\mathbf{q}_{a_i} = [\mathbf{q}_{HAA_i} \quad \mathbf{q}_{HFE_i} \quad \mathbf{q}_{KFE_i}]^T. \quad (5.3)$$

The generalized dynamics model shares the similar formulation as in (2.25) using the QuadISIR structure in Fig. 5.2, and it is rewritten again as below,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}, \quad (5.4)$$

where  $\boldsymbol{\tau} \in \mathbb{R}^{n_r=3n_l}$  is torque for three active actuators and  $\mathbf{S} \in \mathbb{R}^{n_r \times n}$  is the selection matrix which has the different size from those in previous chapters, as follows,

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}_{3 \times 6} & \mathbf{E} & & & & \\ \mathbf{0}_{3 \times 6} & & \mathbf{E} & & & \\ \mathbf{0}_{3 \times 6} & & & \mathbf{E} & & \\ \mathbf{0}_{3 \times 6} & & & & \mathbf{E} & \end{bmatrix}, \quad (5.5)$$

where  $\mathbf{E} = [\mathbf{1}_{3 \times 3} \quad \mathbf{0}_{3 \times 1}] \in \mathbb{R}^{3 \times 4}$ . It means that the actuated torque at each ankle joint is zero. Assuming that there is no slippage between the contact points and the ground, the velocity of the contact points is expressed by the following constraint in (5.6a), and its time derivative is given in (5.6b),

$$\mathbf{v}_c = \mathbf{J}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (5.6a)$$

$$\dot{\mathbf{v}}_c = \mathbf{J}_c \ddot{\mathbf{q}} + \dot{\mathbf{J}}_c \dot{\mathbf{q}} = \mathbf{0}, \quad (5.6b)$$

where  $\mathbf{v}_c$  is the translational velocities of all contact points. In addition, the parallel mechanism induces another two kinematics constraints as follows,

$$\dot{\mathbf{q}}_{KFE} + \dot{\mathbf{q}}_{Ankle} = \mathbf{J}_m \dot{\mathbf{q}} = \mathbf{0}_{4 \times 1}, \quad (5.7a)$$

$$\ddot{\mathbf{q}}_{KFE} + \ddot{\mathbf{q}}_{Ankle} = \mathbf{J}_m \ddot{\mathbf{q}} + \dot{\mathbf{J}}_m \dot{\mathbf{q}} = \mathbf{0}_{4 \times 1}, \quad (5.7b)$$

where  $\dot{\mathbf{q}}_{KFE}$  and  $\dot{\mathbf{q}}_{Ankle}$  combine all the KFE and ankle joints respectively. The Jacobian for this mechanism constraint  $\mathbf{J}_m \in \mathbb{R}^{4 \times n}$  is also one constant selection matrix and its derivative  $\dot{\mathbf{J}}_m = \mathbf{0}$ ,

$$\mathbf{J}_m = \begin{bmatrix} \mathbf{0}_{1 \times 6} & \mathbf{E}_m & & & & \\ \mathbf{0}_{1 \times 6} & & \mathbf{E}_m & & & \\ \mathbf{0}_{1 \times 6} & & & \mathbf{E}_m & & \\ \mathbf{0}_{1 \times 6} & & & & \mathbf{E}_m & \end{bmatrix}, \quad (5.8)$$

where  $\mathbf{E}_m = [0 \ 0 \ 1 \ 1]$ . In addition, the whole-system dynamics model can be defined only depending on the minimum DOFs without the items relating to the ankle joints as follows,

$$\underline{\mathbf{M}} \ddot{\mathbf{q}} + \underline{\mathbf{C}} + \underline{\mathbf{G}} = \bar{\mathbf{J}}_{a|m}^T (\mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}), \quad (5.9)$$

where  $\mathbf{J}_{a|m} = \mathbf{J}_a \mathbf{N}_m$ , and  $\mathbf{J}_a = [\mathbf{J}_b^T \ \mathbf{S}^T]^T \in \mathbb{R}^{(n-4) \times n}$  is like one Jacobian for the joint motion except the ankle joints and  $\mathbf{J}_b = [\mathbf{1}_{6 \times 6} \ \mathbf{0}_{(n-n_b) \times n}]$  denotes the Jacobian for the spatial base motion. Furthermore,  $\mathbf{N}_m = \mathbf{1}_{n \times n} - \bar{\mathbf{J}}_m \mathbf{J}_m$  denotes the null-space projector for the constraints relating to the parallel mechanism, and  $\bar{\mathbf{J}}_m$  is the dynamically consistent inverse of  $\mathbf{J}_m$ , and  $\bar{\mathbf{J}}_m = \mathbf{M}^{-1} \mathbf{J}_m^T (\mathbf{J}_m \mathbf{M}^{-1} \mathbf{J}_m^T)^{-1}$ . In addition,  $\ddot{\mathbf{q}} = \mathbf{J}_{a|m} \ddot{\mathbf{q}} = \mathbf{J}_a \ddot{\mathbf{q}}$  denotes the joint acceleration except the ankle joints. The new inertia matrix  $\underline{\mathbf{M}} \in \mathbb{R}^{(n-4) \times (n-4)}$ , the coriolis and centripetal force  $\underline{\mathbf{C}} \in \mathbb{R}^{(n-4)}$  and the gravity force  $\underline{\mathbf{G}} \in \mathbb{R}^{(n-4)}$  can be derived as below,

$$\underline{\mathbf{M}} = (\mathbf{J}_{a|m} \mathbf{M}^{-1} \mathbf{J}_{a|m}^T)^{-1}, \quad (5.10a)$$

$$\bar{\mathbf{J}}_{a|m}^T = \underline{\mathbf{M}} \mathbf{J}_{a|m} \mathbf{M}^{-1}, \quad (5.10b)$$

$$\underline{\mathbf{C}} = \bar{\mathbf{J}}_{a|m}^T \mathbf{C}, \quad (5.10c)$$

$$\underline{\mathbf{G}} = \bar{\mathbf{J}}_{a|m}^T \mathbf{G}, \quad (5.10d)$$

where  $\mathbf{M}$  and  $\underline{\mathbf{M}}$  are always invertible. The modified model in (5.9) will be used to build the optimization based torque controller in Section 5.4.2.

### 5.2.1.2 New Leg Design:

We took inspiration in the anatomy of quadruped animals: tigers and lions, which have legs constituted of four joints with a parallelogram structure to drive the fourth joint. Actually, the four-joint leg already exists both in the research literature [Hyun 2014], [Zeng 2019], [Kim 2013] and in the business company, e.g. Bigdog developed in Boston Dynamics [Raibert 2008], Cassie in Agility Robotics [Gong 2019]. However, no structure-design principle is proposed or published, especially considering the leg-CoM motion and the relationship with the foot motion. In this section, we will propose one new design principle to express the relationship between the leg CoM motion and the foot motion in a simpler way. The detailed structure and expressions of the four-joint mechanism are detailed and shown in Fig. 5.3. Joints  $i \in \{1, \dots, 4\}$  all rotate around their  $\hat{\mathbf{z}}$  axes and their relative frames are shown in Fig. 5.3.

We will use the structure in Fig. 5.3 to derive our design principle. The real leg  $i$  foot position in the base frame  $\tilde{\mathbf{p}}_{c_i}$  is derived as follows,

$$\tilde{\mathbf{p}}_{c_i} = \mathbf{R}_{i_1} {}^{i_1} \mathbf{p}_{c_i} + {}^b \mathbf{p}_{i_1}, \quad (5.11a)$$

$${}^{i_1} \mathbf{p}_{c_i} = \mathbf{R}_{i_2} {}^{i_4} \mathbf{p}_{c_i} + \mathbf{R}_{i_2} \mathbf{R}_{i_3} {}^{i_3} \mathbf{p}_{i_4} + \mathbf{R}_{i_2} {}^{i_2} \mathbf{p}_{i_3} + {}^{i_1} \mathbf{p}_{i_2}, \quad (5.11b)$$

where  ${}^{i_1} \mathbf{p}_{c_i}$  denotes the foot position in its HAA frame, whose  $x$  component is constant as  $const_1$ .  $\mathbf{R}_{i_1}$  represents the rotation matrix of the leg  $i$  HAA joint relative to the base frame,  $\mathbf{R}_{i_2}$ ,  $\mathbf{R}_{i_3}$  are the ones of

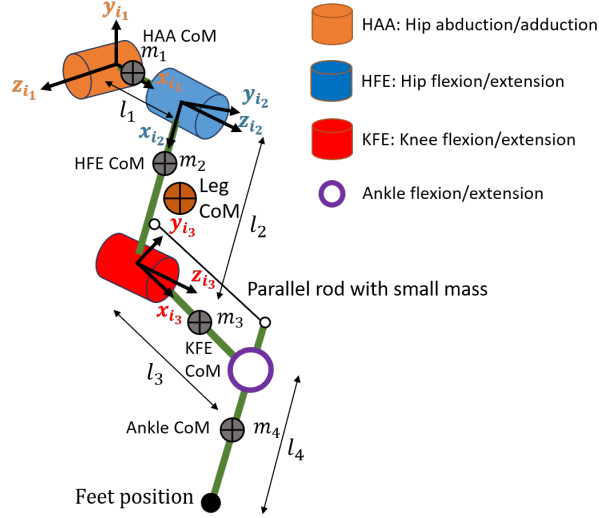


Figure 5.3: New design principle of leg  $i$ . In the sagittal plane, the hip point, leg CoM and the contact point are on the same line.

the frame HFE relative to the HAA frame, and the frame KFE relative to the HFE frame relatively.  $\mathbf{R}_{i_4}$  denotes the orientation of the ankle frame relative to the KFE frame. With the parallelogram structure,  $\mathbf{R}_{i_3}\mathbf{R}_{i_4} = \mathbf{R}_{i_4}\mathbf{R}_{i_3} = \mathbf{1}_{3 \times 3}$ . In addition,  ${}^{i_4}\mathbf{p}_{c_i}$ ,  ${}^{i_3}\mathbf{p}_{i_4}$ ,  ${}^{i_2}\mathbf{p}_{i_3}$ ,  ${}^{i_1}\mathbf{p}_{i_2}$  and  ${}^b\mathbf{p}_{i_1}$  represent respectively the foot position in ankle frame, the ankle joint position in the KFE frame, the KFE joint position in the HFE frame, the HFE joint position in the HAA frame, and HAA joint position in the base frame, which are all constant vectors. The leg CoM position  $\tilde{\mathbf{p}}_i$  in the base frame can be derived as follows,

$$\tilde{\mathbf{p}}_i = \mathbf{R}_{i_1} {}^{i_1}\mathbf{p}_i + {}^b\mathbf{p}_{i_1}, \quad (5.12a)$$

$${}^{i_1}\mathbf{p}_i = \frac{m_1 {}^{i_1}\mathbf{p}_{i_{1cm}} + m_2 {}^{i_1}\mathbf{p}_{i_{2cm}} + m_3 {}^{i_1}\mathbf{p}_{i_{3cm}} + m_4 {}^{i_1}\mathbf{p}_{i_{4cm}}}{m_1 + m_2 + m_3 + m_4}, \quad (5.12b)$$

where  ${}^{i_1}\mathbf{p}_i$  denotes the leg  $i$  CoM position in the HAA frame.  $m_1$ ,  $m_2$ ,  $m_3$  and  $m_4$  represent the masses for the HAA link, the HFE link, the KFE link and the Ankle link respectively.  ${}^{i_1}\mathbf{p}_{i_{1cm}}$ ,  ${}^{i_1}\mathbf{p}_{i_{2cm}}$ ,  ${}^{i_1}\mathbf{p}_{i_{3cm}}$  and  ${}^{i_1}\mathbf{p}_{i_{4cm}}$  denote respectively the CoM positions of the HAA, HFE, KFE and Ankle links in the HAA frame, and they can be derived as below,

$${}^{i_1}\mathbf{p}_{i_{1cm}} = [const_2, 0, 0]^T, \quad (5.13a)$$

$${}^{i_1}\mathbf{p}_{i_{2cm}} = \mathbf{R}_{i_2} {}^{i_2}\mathbf{p}_{i_{2cm}} + {}^{i_1}\mathbf{p}_{i_2}, \quad (5.13b)$$

$${}^{i_1}\mathbf{p}_{i_{3cm}} = \mathbf{R}_{i_2}\mathbf{R}_{i_3} {}^{i_3}\mathbf{p}_{i_{3cm}} + \mathbf{R}_{i_2} {}^{i_2}\mathbf{p}_{i_3} + {}^{i_1}\mathbf{p}_{i_2}, \quad (5.13c)$$

$${}^{i_1}\mathbf{p}_{i_{4cm}} = \mathbf{R}_{i_2} {}^{i_4}\mathbf{p}_{i_{4cm}} + \mathbf{R}_{i_2}\mathbf{R}_{i_3} {}^{i_3}\mathbf{p}_{i_4} + \mathbf{R}_{i_2} {}^{i_2}\mathbf{p}_{i_3} + {}^{i_1}\mathbf{p}_{i_2}, \quad (5.13d)$$

where  $const_2$  denotes one user-defined constant value. To ensure a linear relationship between the feet position and the leg-CoM position in the HAA frame, the relationship can be simply derived by comparing (5.11b) and (5.12b) as follows,

$${}^{i_1}\mathbf{p}_{c_i} = \boldsymbol{\lambda} {}^{i_1}\mathbf{p}_i, \quad (5.14)$$

where  $\boldsymbol{\lambda}$  is a user-defined constant diagonal matrix and  $\boldsymbol{\lambda}$  is the same for all legs as below,

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_x & & \\ & \lambda_y = \frac{\mathbb{K}_y m}{m_2 + m_3 + m_4} & \\ & & \lambda_z = \frac{\mathbb{K}_z m}{m_2 + m_3 + m_4} \end{bmatrix}, \quad (5.15)$$

where  $m = m_1 + m_2 + m_3 + m_4$  represents the leg weight. Note that the four legs have the same weight.  $\mathbb{K}_y$  and  $\mathbb{K}_z$  are dependent on the mechanical design. Substituting (5.11b) and (5.12b) into (5.14), we

Table 5.1: Parameters design of one leg:  $l_1 = 121.5mm$ ,  $l_2 = 250mm$ ,  $l_3 = 200mm$ ,  $l_4 = 250mm$  in Fig. 5.3.

mass(kg)	Joint position(mm)	CoM position(mm)
$m_1 = 1.6$	${}^{i_1}\mathbf{p}_{i_2} = [l_1, 0, 0]^T$	${}^{i_1}\mathbf{p}_{i_{1cm}} = [\frac{1}{3}l_1, 0, 0]^T$
$m_2 = 3.6$	${}^{i_2}\mathbf{p}_{i_3} = [l_2, 0, 0]^T$	${}^{i_2}\mathbf{p}_{i_{2cm}} = [\frac{1}{3}l_2, 0, 0]^T$
$m_3 = 1.2$	${}^{i_3}\mathbf{p}_{i_4} = [l_3, 0, 0]^T$	${}^{i_3}\mathbf{p}_{i_{3cm}} = [\frac{2}{3}l_3, 0, 0]^T$
$m_4 = 1.2$	${}^{i_4}\mathbf{p}_{c_i} = [l_4, 0, 0]^T$	${}^{i_4}\mathbf{p}_{i_{4cm}} = [\frac{1}{3}l_4, 0, 0]^T$

can get the detailed relationships as follows,

$$\lambda_x = \frac{m}{m_2 + m_3 + m_4 + m_1 \frac{const_2}{const_1}}, \quad (5.16a)$$

$${}^{i_4}\mathbf{p}_{c_i} + {}^{i_2}\mathbf{p}_{i_3} = \mathbb{K}_y \frac{m_2 {}^{i_2}\mathbf{p}_{i_{2cm}} + m_3 {}^{i_2}\mathbf{p}_{i_3} + m_4 ({}^{i_4}\mathbf{p}_{i_{4cm}} + {}^{i_2}\mathbf{p}_{i_3})}{m_2 + m_3 + m_4}, \quad (5.16b)$$

$${}^{i_3}\mathbf{p}_{i_4} = \mathbb{K}_z \frac{m_3 {}^{i_3}\mathbf{p}_{i_{3cm}} + m_4 {}^{i_3}\mathbf{p}_{i_4}}{m_2 + m_3 + m_4}, \quad (5.16c)$$

where  $\lambda_x$  is constant. In the design process, when (5.16) is satisfied, we can achieve the simple relationship between the CoM and feet positions in the HAA frame, shown in (5.14). For our robot, with suitable mass distribution and link length design of each leg body, shown in Table. 5.1, we achieve  $\mathbb{K}_y = \mathbb{K}_z = 3$ , therefore,  $\boldsymbol{\lambda} \approx \text{diag}[1.164, 3.8, 3.8]$  which satisfies that the leg CoM should be close to the hip [Wensing 2017b]. Then the simple relationship between the leg CoM and the leg contact positions in the base frame is achieved in the following equation by combining (5.11a), (5.12a) and (5.14),

$$\tilde{\mathbf{p}}_{c_i} = \underbrace{\mathbf{R}_{i_1} \boldsymbol{\lambda} \mathbf{R}_{i_1}^T}_{\boldsymbol{\lambda}_i} (\tilde{\mathbf{p}}_i - {}^b\mathbf{p}_{i_1}) + {}^b\mathbf{p}_{i_1}, \quad (5.17)$$

where  $\boldsymbol{\lambda}_i$  only depends on the leg  $i$  HAA joint position,  $\mathbf{q}_{HAA_i}$ , as follows,

$$\boldsymbol{\lambda}_i = \begin{bmatrix} \lambda_z & 0 & 0 \\ 0 & \lambda_x \cos^2 \mathbf{q}_{HAA_i} + \lambda_y \sin^2 \mathbf{q}_{HAA_i} & (\lambda_x - \lambda_y) \cos \mathbf{q}_{HAA_i} \sin \mathbf{q}_{HAA_i} \\ 0 & (\lambda_x - \lambda_y) \cos \mathbf{q}_{HAA_i} \sin \mathbf{q}_{HAA_i} & \lambda_x \cos^2 \mathbf{q}_{HAA_i} + \lambda_y \sin^2 \mathbf{q}_{HAA_i} \end{bmatrix}, \quad (5.18)$$

It is noticed that as one of the leg CoM and contact positions is known, the other one can be calculated out directly. This simple relationship has more advantages:

- Since the feet positions are related to the leg-CoM position, this relationship enables us to control each leg centroidal motions, instead of controlling each leg-foot motion. This presents a potential for generating the leg trajectories without leg-massless assumptions.
- It enables the robot to trot with symmetrical contact position and with symmetrical dynamic properties.
- When the specifications related to the robot centroidal momentum/dynamics should be controlled, the method can be used to generate suitable legged motion to track the centroidal motion reference more intuitively.
- We can use the virtual joint motion (between the base and each leg CoM) to build the centroidal momentum/dynamics model, which is much easier than using each leg-joint motion.
- Trajectory optimization can also take advantage of this relationship, providing fewer optimization variables

In addition, this simple relationship in (5.17) can be further simplified as:  $\tilde{\mathbf{p}}_{c_i} = \mathbf{K}(\tilde{\mathbf{p}}_i - {}^b\mathbf{p}_{i_1}) + {}^b\mathbf{p}_{i_1}$ , if one spherical actuator is applied for the HAA and HFE joints. This kind of actuator can rotate in two or three axes independently [Kumar 2019], which means that the origins of the HAA and HFE joints can be defined to be coincident. However, since so far, such actuators are still in the developing process in the research field.

### 5.2.1.3 Actuated Motion:

The actuated-joint accelerations can be derived by using the actuated torque and the modified dynamics model in (5.9) as follows,

$$\underline{\ddot{\mathbf{q}}} = \underline{\mathbf{M}}^{-1} \left[ \bar{\mathbf{J}}_{a|m}^T (\mathbf{J}_c^T \mathbf{F}_c + \mathbf{S}^T \boldsymbol{\tau}) - \underline{\mathbf{C}} - \underline{\mathbf{G}} \right]. \quad (5.19)$$

The contact force  $\mathbf{F}_c$  can be estimated by using the generalized dynamics model in (5.4) and the contact constraints in (5.6) as below,

$$\mathbf{F}_c = -\bar{\mathbf{J}}_{c|m}^T \mathbf{S}^T \boldsymbol{\tau} + \boldsymbol{\mu}_{c|m} + \boldsymbol{\rho}_{c|m}, \quad (5.20)$$

where  $\mathbf{J}_{c|m} = \mathbf{J}_c \mathbf{N}_m$ . In addition,  $\boldsymbol{\mu}_{c|m}$  and  $\boldsymbol{\rho}_{c|m}$  denote the operational-space coriolis and centrifugal force, and the gravity force at the contact points, and they are derived to be in the null-space of the constraints of the parallel mechanism as follows,

$$\boldsymbol{\Lambda}_{c|m} = (\mathbf{J}_{c|m} \mathbf{M}^{-1} \mathbf{J}_{c|m}^T)^{-1}, \quad (5.21a)$$

$$\bar{\mathbf{J}}_{c|m}^T = \boldsymbol{\Lambda}_{c|m} \mathbf{J}_{c|m} \mathbf{M}^{-1}, \quad (5.21b)$$

$$\boldsymbol{\mu}_{c|m} = \bar{\mathbf{J}}_{c|m}^T \mathbf{C} - \boldsymbol{\Lambda}_{c|m} \dot{\mathbf{J}}_c \dot{\mathbf{q}}, \quad (5.21c)$$

$$\boldsymbol{\rho}_{c|m} = \bar{\mathbf{J}}_{c|m}^T \mathbf{G}. \quad (5.21d)$$

where  $\boldsymbol{\Lambda}_{c|m}$  represents the operational-space inertia at the contact points which is projected into the null-space of the parallel kinematic constraint. 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.

## 5.2.2 Each Leg Centroidal Motion

In this chapter, we control the quadruped centroidal motion which consists of the base and each leg motion. We plan to control each leg CoM motion in the operational space, instead of each joint motion in this leg. Therefore, we derive each leg-CoM position, velocity and acceleration and its relative Jacobian in this section.

### 5.2.2.1 Leg CoM Position and Jacobian:

To integrate the task of one leg CoM into the operational space control, the Jacobian at leg  $i$  CoM  $\mathbf{J}_i \in \mathbb{R}^{6 \times n}$  can be derived as below,

$$\mathbf{J}_i = \mathbf{I}_i^{-1} ({}^b\mathbf{X}_i^T \mathbf{U} \mathbf{M}_i) = [\mathbf{J}_i^b \quad \mathbf{J}_i^p], \quad (5.22)$$

where  $\mathbf{U} = [\mathbf{1}_{6 \times 6}, \mathbf{0}_{6 \times (n-6)}]$ .  $\mathbf{M}_i \in \mathbb{R}^{n \times n}$  is the generalized inertia matrix (different with  $\mathbf{M}$ ), and it is achieved by assuming that in the whole system, only the body inertias in leg  $i$  are nonzero.  $\mathbf{J}_i^b \in \mathbb{R}^{n_b \times n_b}$  relates to the left  $n_b$  columns and  $\mathbf{J}_i^p \in \mathbb{R}^{6 \times (n-n_b)}$  represents the right  $(n-n_b)$  columns of  $\mathbf{J}_i$ . The leg  $i$  CoM composite inertia is derived as follows,

$$\mathbf{I}_i = {}^b\mathbf{X}_i^T (\mathbf{U} \mathbf{M}_i \mathbf{U}^T) {}^b\mathbf{X}_i = \begin{bmatrix} \bar{\mathbf{I}}_i & \\ & \mathbf{m} \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad (5.23)$$

where  $\bar{\mathbf{I}}_i$  is the angular inertia at the leg  $i$  CoM and  $\mathbf{m} = m \mathbf{1}_{3 \times 3}$  is the translational inertia. The transformation matrix  ${}^b\mathbf{X}_i$  (from the leg CoM frame to the base frame) can be achieved by setting the

leg CoM frame parallel with the base frame as,

$${}^b\mathbf{X}_i = \begin{bmatrix} {}^b\mathbf{R}_i & \mathbf{0} \\ \tilde{\mathbf{p}}_i \times {}^b\mathbf{R}_i & {}^b\mathbf{R}_i \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \tilde{\mathbf{p}}_i \times & \mathbf{1} \end{bmatrix}, \quad (5.24)$$

where  $\tilde{\mathbf{p}}_i$  denotes the leg  $i$  CoM position in the base frame and can be derived by  $\mathbf{M}_i$  as below,

$$\tilde{\mathbf{p}}_i = \frac{[(\mathbf{UM}_i\mathbf{U}^T)_{(3,5)}, (\mathbf{UM}_i\mathbf{U}^T)_{(1,6)}, (\mathbf{UM}_i\mathbf{U}^T)_{(2,4)}]^T}{m}, \quad (5.25)$$

where the subscript  $(i, j)$  denotes the component at row  $i$  and column  $j$  of one matrix.

In addition, the bias force  $\dot{\mathbf{J}}_i\dot{\mathbf{q}}$  can be derived as follows,

$$\dot{\mathbf{J}}_i\dot{\mathbf{q}} = -\mathbf{I}_i^{-1}(\mathbf{v}_i \times \mathbf{I}_i\mathbf{v}_i - {}^b\mathbf{X}_i^T\mathbf{UC}_i), \quad (5.26)$$

where  $\mathbf{v}_i$  denotes the leg  $i$  CoM spatial velocity.  $\mathbf{C}_i$  is derived with  $\mathbf{C}$  by using the Recursively Newton-Euler Algorithm [Luh 1980]. It is noticed that  $\mathbf{C}_i$  is different with  $\mathbf{C}$  by assuming that in the whole system, only the body inertias in leg  $i$  are nonzero.

### 5.2.2.2 Leg CoM Velocity and Acceleration:

The leg  $i$  CoM spatial velocity  $\mathbf{v}_i$  and acceleration  $\dot{\mathbf{v}}_i$  can also be expressed as follows,

$$\begin{aligned} \mathbf{v}_i &= \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} = \mathbf{J}_i\dot{\mathbf{q}} = \mathbf{J}_i^b\mathbf{v}_b + \mathbf{J}_i^a\dot{\mathbf{q}}_{a_i}, \\ &= {}^i\mathbf{X}_b\mathbf{v}_b + \tilde{\mathbf{v}}_i = {}^i\mathbf{X}_b \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_b \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{\omega}}_i \\ \tilde{\mathbf{v}}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_b + \tilde{\boldsymbol{\omega}}_i \\ \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + \mathbf{v}_b + \tilde{\mathbf{v}}_i \end{bmatrix}, \end{aligned} \quad (5.27a)$$

$$\begin{aligned} \dot{\mathbf{v}}_i &= \begin{bmatrix} \dot{\boldsymbol{\omega}}_i \\ \dot{\mathbf{v}}_i \end{bmatrix} = \mathbf{J}_i\ddot{\mathbf{q}} + \dot{\mathbf{J}}_i\dot{\mathbf{q}} = \mathbf{J}_i^b\dot{\mathbf{v}}_b + \mathbf{J}_i^a\ddot{\mathbf{q}}_{a_i} + \dot{\mathbf{J}}_i\dot{\mathbf{q}}, \\ &= {}^i\mathbf{X}_b\dot{\mathbf{v}}_b + \dot{\tilde{\mathbf{v}}}_i + \mathbf{v}_i \times \tilde{\mathbf{v}}_i = \begin{bmatrix} \dot{\boldsymbol{\omega}}_b + \dot{\tilde{\boldsymbol{\omega}}}_i + \boldsymbol{\omega}_b \times \tilde{\boldsymbol{\omega}}_i \\ \dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_i + \dot{\mathbf{v}}_b + \dot{\tilde{\mathbf{v}}}_i + \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + \mathbf{v}_i \times \tilde{\boldsymbol{\omega}}_i \end{bmatrix}, \end{aligned} \quad (5.27b)$$

where  $\mathbf{J}_{c_i}^a \in \mathbb{R}^{6 \times 3}$  corresponds to the leg  $i$  actuated-joint motion.  $\boldsymbol{\omega}_i$  and  $\mathbf{v}_i$  denote the leg  $i$  CoM angular and translational velocities respectively.  $\tilde{\mathbf{v}}_i = [\tilde{\boldsymbol{\omega}}_i \ \tilde{\mathbf{v}}_i]$  is the spatial leg  $i$  velocity relative to the base, in which  $\tilde{\boldsymbol{\omega}}_i$  and  $\tilde{\mathbf{v}}_i$  denote the angular and translational components respectively. It is noticed that  $\tilde{\mathbf{v}}_i = \dot{\tilde{\mathbf{p}}}_i$  and  $\dot{\tilde{\mathbf{v}}}_i = \dot{\tilde{\mathbf{p}}}_i - \tilde{\boldsymbol{\omega}}_i \times \tilde{\mathbf{v}}_i$ . The cross product for spatial vectors are detailed in [Featherstone 2014]. These relationships will be used to achieve the swing-leg CoM motion in Section 5.3. It is noticed that the virtual leg CoM joint has only 3 DOFs, therefore,  $\tilde{\boldsymbol{\omega}}_i$  and  $\tilde{\mathbf{v}}_i$  can be derived from each other. It means that one of  $\tilde{\boldsymbol{\omega}}_i$  and  $\tilde{\mathbf{v}}_i$  is known, the other can be calculated directly. The relationship can be derived as follows,

$$\dot{\mathbf{q}}_{a_i} = (\mathbf{S}_c\mathbf{J}_i^a)^{-1}\tilde{\mathbf{v}}_i = (\mathbf{S}_u\mathbf{J}_i^a)^{-1}\tilde{\boldsymbol{\omega}}_i, \quad (5.28a)$$

$$\ddot{\mathbf{q}}_{a_i} = (\mathbf{S}_c\mathbf{J}_i^a)^{-1}(\dot{\tilde{\mathbf{v}}}_i - \mathbf{S}_c\dot{\mathbf{J}}_i^a\dot{\mathbf{q}}_{a_i}) = (\mathbf{S}_u\mathbf{J}_i^a)^{-1}(\dot{\tilde{\boldsymbol{\omega}}}_i - \mathbf{S}_u\dot{\mathbf{J}}_i^a\dot{\mathbf{q}}_{a_i}), \quad (5.28b)$$

where  $\mathbf{S}_u = [\mathbf{1}_{3 \times 3} \ \mathbf{0}_{3 \times 3}]$  and  $\dot{\mathbf{J}}_i^a\dot{\mathbf{q}}_{a_i}$  can be calculated using the same equation as in (5.26) but assuming the base spatial velocity to be zero. Therefore,  $\tilde{\boldsymbol{\omega}}_i$  and  $\dot{\tilde{\boldsymbol{\omega}}}_i$  can be expressed by  $\dot{\tilde{\mathbf{p}}}_i$  and  $\ddot{\tilde{\mathbf{p}}}_i$  as below,

$$\tilde{\boldsymbol{\omega}}_i = \mathbf{L}_i\dot{\tilde{\mathbf{p}}}_i, \quad (5.29a)$$

$$\dot{\tilde{\boldsymbol{\omega}}}_i = \mathbf{L}_i\ddot{\tilde{\mathbf{p}}}_i - \mathbf{Q}_i, \quad (5.29b)$$

$$\mathbf{L}_i = (\mathbf{S}_u\mathbf{J}_i^a)(\mathbf{S}_c\mathbf{J}_i^a)^{-1}, \quad (5.29c)$$

$$\mathbf{Q}_i = \mathbf{L}_i(\tilde{\boldsymbol{\omega}}_i \times \tilde{\mathbf{v}}_i + \mathbf{S}_c\dot{\mathbf{J}}_i^a\dot{\mathbf{q}}_{a_i}) - \mathbf{S}_u\dot{\mathbf{J}}_i^a\dot{\mathbf{q}}_{a_i}, \quad (5.29d)$$

These relationships (5.29) will be used in the swing-leg and centroidal angular motion generations in Section 5.3.10.



For the stance legs, we can use the contact constraints to express  $\dot{\mathbf{q}}_{a_i}$  and  $\ddot{\mathbf{q}}_{a_i}$ . Given the desired base spatial velocity  $\mathbf{v}_b$ , the actual robot configuration, and the desired actuated joint velocity of the leg  $\dot{\mathbf{q}}_{a_i}$ , then the leg-CoM velocity w.r.t. the base frame  $\tilde{\boldsymbol{\omega}}_i$ ,  $\tilde{\mathbf{v}}_i$  and  $\tilde{\mathbf{p}}_i$  can be calculated. Given the desired base spatial acceleration  $\dot{\mathbf{v}}_b$  and the actual robot configuration/velocity, and the desired actuated joint acceleration of the stance leg  $\ddot{\mathbf{q}}_{a_i}$ , the leg-CoM acceleration w.r.t. the base frame  $\dot{\tilde{\boldsymbol{\omega}}}_i$ ,  $\dot{\tilde{\mathbf{v}}}_i$  and  $\dot{\tilde{\mathbf{p}}}_i$  can be expressed. For the swing legs, we can express  $(\dot{\tilde{\mathbf{p}}}_i, \ddot{\tilde{\mathbf{p}}}_i)$  first, and then use them to obtain  $\dot{\mathbf{q}}_{a_i}$  and  $\ddot{\mathbf{q}}_{a_i}$ . Then  $(\tilde{\boldsymbol{\omega}}_i, \dot{\tilde{\boldsymbol{\omega}}}_i)$  can be achieved as for the stance legs.

### 5.2.2.3 Each Stance-Leg Properties:

Each contact-point velocity and acceleration can be rewritten as follows,

$$\mathbf{v}_{c_i} = \mathbf{J}_{c_i} \dot{\mathbf{q}} = \mathbf{J}_{c_i}^b \mathbf{v}_b + \mathbf{J}_{c_i}^a \dot{\mathbf{q}}_{a_i} = \mathbf{0}_{3 \times 1}, \quad (5.30a)$$

$$\dot{\mathbf{v}}_{c_i} = \mathbf{J}_{c_i} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{c_i} \dot{\mathbf{q}} = \mathbf{J}_{c_i}^b \dot{\mathbf{v}}_b + \mathbf{J}_{c_i}^a \ddot{\mathbf{q}}_{a_i} + \dot{\mathbf{J}}_{c_i} \dot{\mathbf{q}} = \mathbf{0}_{3 \times 1}, \quad (5.30b)$$

where  $\mathbf{v}_b = [\boldsymbol{\omega}_b^T \ \mathbf{v}_b^T]^T$  is the base spatial velocity.  $\boldsymbol{\omega}_b$  and  $\mathbf{v}_b$  denote the base angular and translational velocities respectively.  $\mathbf{J}_{c_i}^b \in \mathbb{R}^{3 \times 6}$  relates to the floating-base motion.  $\mathbf{J}_{c_i}^a \in \mathbb{R}^{3 \times 3}$  corresponds to the leg  $i$  actuated-joint motion. It is noticed that the Jacobian relating to the ankle joint motion is integrated into  $\mathbf{J}_{c_i}^a$  using the property of the parallel mechanism. By given the desired base spatial velocity  $\mathbf{v}_b$  and the actual robot configuration, the desired actuated joint velocity of the stance leg  $i$ ,  $\dot{\mathbf{q}}_{a_i}$  can be achieved. Then we can use (5.28) and (5.29) to achieve the stance-leg  $i$  angular and translational motion in the base frame, including  $(\tilde{\boldsymbol{\omega}}_i, \dot{\tilde{\boldsymbol{\omega}}}_i)$  and  $(\tilde{\mathbf{p}}_i, \dot{\tilde{\mathbf{p}}}_i)$ . Similarly, by given the desired base spatial acceleration  $\dot{\mathbf{v}}_b$  and the actual robot configuration/velocity, the desired actuated joint acceleration of the stance leg  $i$ ,  $\ddot{\mathbf{q}}_{a_i}$  can be achieved.

### 5.2.3 Whole-body Centroidal Jacobian

In this section, we will decompose the robot centroidal motion using each leg-CoM motion instead of real leg bodies. Initially, the robot centroidal position  $\mathbf{p}_G$  can be decomposed by the base and each leg-CoM positions as,

$$\mathbf{p}_G = {}^b \mathbf{R}_G^T \tilde{\mathbf{p}}_G + \mathbf{p}_b, \quad (5.31a)$$

$$\tilde{\mathbf{p}}_G = \frac{m}{m_G} \sum_{i=1}^4 \tilde{\mathbf{p}}_i, \quad (5.31b)$$

where  $\mathbf{p}_G$  and  $\tilde{\mathbf{p}}_G$  represent the robot CoM position in the inertia frame and in the base frame respectively.  $m_G = m_b + 4m$  is the whole-robot mass and  $m_b$  is the base mass.  ${}^b \mathbf{R}_G$  is the rotation matrix from the CoM frame to the base frame. We set the CoM frame parallel with the inertial frame, and  ${}^b \mathbf{R}_G = {}^b \mathbf{R}_0$ . In this way, it is convenient to control the centroidal motion in the inertial frame directly. Even though, we keep the form  ${}^b \mathbf{R}_G$  in the following sections to enable the readers to stick with the user-defined CoM frame. To integrate tasks (relating to the robot CoM) into the operational space controller, its Jacobian  $\mathbf{J}_G$  can be derived as below,

$$\mathbf{J}_G = \mathbf{I}_G^{-1} ({}^b \mathbf{X}_G^T \mathbf{U} \mathbf{M}) \in \mathbb{R}^{6 \times n}, \quad (5.32a)$$

$${}^b \mathbf{X}_G = \begin{bmatrix} {}^b \mathbf{R}_G & \mathbf{0} \\ \tilde{\mathbf{p}}_G \times {}^b \mathbf{R}_G & {}^b \mathbf{R}_G \end{bmatrix}, \quad (5.32b)$$

where  $\mathbf{I}_G^C$  is the robot composite inertia at the base.  ${}^b \mathbf{X}_G \in \mathbb{R}^{6 \times 6}$  is the transformation matrix for spatial vectors from the CoM frame to the base frame.  $\mathbf{I}_G$  denotes the composite rigid body inertia at the

quadruped CoM [Siciliano 2016] as follows,

$$\mathbf{I}_G = {}^b\mathbf{X}_G^T \mathbf{I}_b^C {}^b\mathbf{X}_G = \begin{bmatrix} \bar{\mathbf{I}}_G & \\ & \mathbf{m}_G \end{bmatrix}, \quad (5.33a)$$

$$\mathbf{I}_b^C = \mathbf{U} \mathbf{M} \mathbf{U}^T = \begin{bmatrix} \bar{\mathbf{I}}_b^C & m_G \tilde{\mathbf{p}}_G \times \\ -m_G \tilde{\mathbf{p}}_G \times & m_G \mathbf{1}_{3 \times 3} \end{bmatrix}, \quad (5.33b)$$

where  $\bar{\mathbf{I}}_G$  and  $\mathbf{m}_G = m_G \mathbf{1}_{1 \times 3}$  represent the robot centroidal angular and translational inertias respectively. The robot CoM spatial velocity is derived easily as  $\mathbf{v}_G = [\boldsymbol{\omega}_G^T \mathbf{v}_G^T]^T = \mathbf{J}_G \dot{\mathbf{q}}$  in which  $\boldsymbol{\omega}_G$  and  $\mathbf{v}_G$  represent the centroidal angular and translational velocities respectively. The centroidal position in the base frame can also be derived using (5.33b) as below,

$$\tilde{\mathbf{p}}_G = \frac{1}{m_G} \left[ \mathbf{I}_{b(3,5)}^C \quad \mathbf{I}_{b(1,6)}^C \quad \mathbf{I}_{b(2,4)}^C \right]^T. \quad (5.34)$$

It is noticed that the operational space inertia at the robot CoM,  $\boldsymbol{\Lambda}_G = (\mathbf{J}_G \mathbf{M}^{-1} \mathbf{J}_G^T)^{-1}$  is equal with  $\mathbf{I}_G$ . However, for the operational space inertia at leg  $i$  CoM,  $\boldsymbol{\Lambda}_i = (\mathbf{J}_i \mathbf{M}_i^+ \mathbf{J}_i^T)^{-1}$  is not equal with  $\mathbf{I}_i$ , in which  $+$  denotes the Moore-Penrose inverse.

#### 5.2.4 Centroidal Model Decompositions

To prepare for the new legged locomotion in Section 5.3, we derive the centroidal momentum/dynamics model using parameters of each leg CoM instead of the real leg bodies. The robot centroidal momentum is rewritten by combining parameters of the base and each leg CoM as below,

$$\underbrace{\mathbf{I}_G \mathbf{v}_G}_{\mathbf{h}_G} = \begin{bmatrix} \mathbf{k}_G \\ \mathbf{l}_G \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_G \boldsymbol{\omega}_G \\ \mathbf{m}_G \mathbf{v}_G \end{bmatrix} = \underbrace{{}^b\mathbf{X}_G^T \mathbf{I}_b \mathbf{v}_b}_{\mathbf{h}_b} + \sum_{i=1}^4 \underbrace{{}^i\mathbf{X}_G^T \mathbf{I}_i \mathbf{v}_i}_{\mathbf{h}_i}, \quad (5.35)$$

where  $\mathbf{h}_G = [\mathbf{k}_G^T \quad \mathbf{l}_G^T]^T$  denotes the robot centroidal momentum.  $\mathbf{k}_G$  and  $\mathbf{l}_G$  denote the angular and translational components respectively.  $\mathbf{h}_b$  and  $\mathbf{h}_i$  represent the contributed centroidal momentum relating to the base and leg  $i$  respectively.  $\mathbf{I}_b \in \mathbb{R}^{6 \times 6}$  represents the base inertia and it is a diagonal block matrix when the base frame is coincident with its CoM as follows,

$$\mathbf{I}_b = \begin{bmatrix} \bar{\mathbf{I}}_b & \\ & \mathbf{m}_b \end{bmatrix}, \quad \mathbf{m}_b = m_b \mathbf{1}_{3 \times 3}, \quad (5.36)$$

where  $\bar{\mathbf{I}}_b$  and  $\mathbf{m}_b$  denote the angular and translational inertias of the floating base respectively.  ${}^i\mathbf{X}_G \in \mathbb{R}^{6 \times 6}$  in (5.35) is the transformation matrix for spatial vectors from the CoM frame to the leg  $i$  CoM frame as below,

$${}^i\mathbf{X}_G = \begin{bmatrix} {}^i\mathbf{R}_G & \mathbf{0} \\ {}^i\mathbf{p}_G \times {}^i\mathbf{R}_G & {}^i\mathbf{R}_G \end{bmatrix} = \begin{bmatrix} {}^b\mathbf{R}_G & \mathbf{0} \\ {}^i\mathbf{p}_G \times {}^b\mathbf{R}_G & {}^b\mathbf{R}_G \end{bmatrix}, \quad (5.37)$$

where  ${}^i\mathbf{p}_G$  is the centroidal position relative to the leg  $i$  CoM as follows,

$${}^i\mathbf{p}_G = \tilde{\mathbf{p}}_G - \tilde{\mathbf{p}}_i. \quad (5.38)$$

The angular and translational centroidal momentums can be rewritten based on the base spatial motion and each leg-CoM motion as below,

$$\mathbf{k}_G = {}^b\mathbf{R}_G^T \left\{ \bar{\mathbf{I}}_b \boldsymbol{\omega}_b + \sum_{i=1}^4 \left[ \bar{\mathbf{I}}_i \boldsymbol{\omega}_i - {}^i\mathbf{p}_G \times \mathbf{m} (\boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + \dot{\tilde{\mathbf{p}}}_i) \right] \right\}, \quad (5.39a)$$

$$\mathbf{l}_G = {}^b\mathbf{R}_G^T \left[ \mathbf{m}_b \mathbf{v}_b + \mathbf{m} \sum_{i=1}^4 (\boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + \mathbf{v}_b + \dot{\tilde{\mathbf{p}}}_i) \right]. \quad (5.39b)$$

The centroidal dynamics is rewritten as follows,

$$\begin{aligned}\dot{\mathbf{h}}_G &= \mathbf{I}_G \dot{\mathbf{v}}_G + \mathbf{v}_G \times \mathbf{I}_G \mathbf{v}_G, \\ &= \begin{bmatrix} \dot{\mathbf{k}}_G \\ \dot{\mathbf{l}}_G \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_G \dot{\boldsymbol{\omega}}_G + \boldsymbol{\omega}_G \times \bar{\mathbf{I}}_G \boldsymbol{\omega}_G \\ \mathbf{m}_G (\dot{\mathbf{v}}_G + \boldsymbol{\omega}_G \times \mathbf{v}_G) \end{bmatrix} = \dot{\mathbf{h}}_b + \sum_{i=1}^4 \dot{\mathbf{h}}_i,\end{aligned}\quad (5.40)$$

where  $\dot{\mathbf{h}}_b$  and  $\dot{\mathbf{h}}_i$  are achieved as below,

$$\dot{\mathbf{h}}_b = {}^b \mathbf{X}_G^T (\mathbf{I}_b \dot{\mathbf{v}}_b + \mathbf{v}_b \times \mathbf{I}_b \mathbf{v}_b), \quad (5.41a)$$

$$\dot{\mathbf{h}}_i = {}^i \mathbf{X}_G^T (\mathbf{I}_i \dot{\mathbf{v}}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i). \quad (5.41b)$$

The angular and translational centroidal dynamics are rewritten as below,

$$\dot{\mathbf{k}}_G = {}^b \mathbf{R}_G^T (\bar{\mathbf{I}}_b \dot{\boldsymbol{\omega}}_b + \mathbf{e}) + {}^b \mathbf{R}_G^T \sum_{i=1}^4 [\bar{\mathbf{I}}_i \dot{\boldsymbol{\omega}}_i - {}^i \mathbf{p}_G \times \mathbf{m} (\dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_i + \ddot{\mathbf{p}}_i)], \quad (5.42a)$$

$$\dot{\mathbf{l}}_G = {}^b \mathbf{R}_G^T \left[ \mathbf{m}_b \dot{\mathbf{v}}_b + \mathbf{b} + \mathbf{m} \sum_{i=1}^4 (\dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_i + \dot{\mathbf{v}}_b + \ddot{\mathbf{p}}_i) \right], \quad (5.42b)$$

where  $\mathbf{e}$  and  $\mathbf{b}$  only depend on the body velocities as below,

$$\mathbf{e} = \boldsymbol{\omega}_b \times \bar{\mathbf{I}}_b \boldsymbol{\omega}_b + \sum_{i=1}^4 \boldsymbol{\omega}_i \times \bar{\mathbf{I}}_i \boldsymbol{\omega}_i - \sum_{i=1}^4 \{ {}^i \mathbf{p}_G \times [\boldsymbol{\omega}_b \times \mathbf{m} (\boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + 2\dot{\tilde{\mathbf{p}}}_i)] \}, \quad (5.43a)$$

$$\mathbf{b} = \boldsymbol{\omega}_b \times \mathbf{m}_b \mathbf{v}_b + \mathbf{m} \sum_{i=1}^4 (\boldsymbol{\omega}_b \times \dot{\tilde{\mathbf{p}}}_i + \boldsymbol{\omega}_b \times \mathbf{v}_i). \quad (5.43b)$$

The centroidal motion decompositions in (5.31), (5.39) and (5.42) serve as the basis to generate the swing-leg positions, velocities and accelerations respectively in Section 5.3.

## 5.3 Legged Motion Generation

In this section, we propose a new general legged locomotion principle for the quadruped trotting and walking gait. Inspired from nature, we believe that the legged motion is generated by considering the quadruped whole-body dynamics properties. Therefore, the new principle is developed by using the robot centroidal momentum/dynamics model in Section 5.2.4. In addition, the principle should enable the robot to run in the environment without defining the internal legged motions, which is similar with vehicles for which the wheel motion can be generated on-line from the motion references of the vehicle base. Similarly, the legged motion should be consistent with the motion reference of the quadruped floating base. Our method enables the robot to generate the walking and trotting gaits automatically without defining specific legged trajectories. In this chapter, we concentrate on the locomotion modes without flight phases which will be handled by passive dynamics in future works.

### 5.3.1 New Quadruped Locomotion Principle

Inspired by human and quadruped animals, we just consider our whole-body composite motion (e.g. centroidal motion). It is not necessary to think how we walk, especially the stance-leg motion, which means that we can generate the legged locomotion automatically without thinking too much about each-leg detailed trajectories.

We give one simple example shown in Fig. 5.4: a two-leg robot runs in the sagittal plane. The robot centroidal motion reference is generated by the base and two leg motion. We define the front leg as the swing leg and the hind leg as the stance leg. When the robot runs in the forward direction, we can simply set the base forward motion to be same as the centroidal forward motion. The stance leg

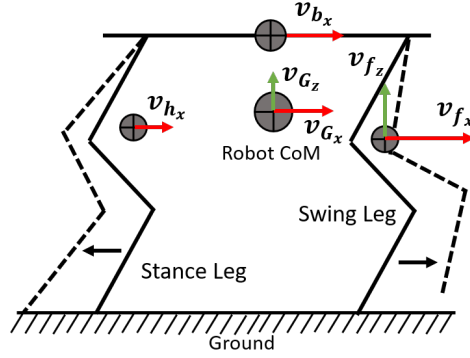


Figure 5.4: An example in the sagittal plane to show our new locomotion principle using two legs for tracking the robot centroidal motion.  $v_f$  and  $v_h$  denote the velocity of the front and hind leg CoMs respectively.  $x$  and  $z$  represent the front and vertical directions respectively.

will just follow the base motion without conflicting the contact constraints. The stance leg will delay backwards compared to the desired centroidal motion. Therefore, to track the centroidal forward motion, the front-swing leg should advance the base to compensate the stance-leg delay influence. As shown in Fig. 5.4, the length of the red arrows show the whole-body forward motion relationship. To enable the swing legs to lift off, the centroidal vertical motion reference should be defined with bump-like shapes. In this way, with the suitable centroidal motion design, the front leg can generate the lift-off motion automatically. The green arrows in Fig. 5.4 show the whole-body vertical motion relationship.

This legged motion approach is simple and is closer to the locomotion principle of real quadruped animals. The all leg trajectories can be generated on-line automatically. This principle can also handle the lateral large-force push recovery motion.

This approach is not limited to the trotting and walking gaits, but it can also be extended to various gaits with compatible centroidal and base motion trajectories, which is much closer to the real legged locomotion principle for the quadruped animals.

### 5.3.2 New Legged Motion Generator

The legged motion generator is established in Fig. 5.5 and it is based on the new locomotion principle. The motion references include two levels:

- (1) External level: the base translational motion in the inertial frame.
- (2) Internal level: the base angular motion, and the centroidal translational motion relative to the base.

In the external level, the base translational motion is defined or can be generated using optimization based techniques, e.g. model based predictive control.

The internal motion references include the centroidal translational motion relative to the base, and the base angular motion. The centroidal angular motion references are achieved only for the trotting mode. The relative translational motion between the robot CoM and the base motions is defined as follows,

$$\mathbf{p}_{Gb} = \mathbf{p}_G - \mathbf{p}_b = \begin{bmatrix} \mathbf{p}_{Gb_x} \\ \mathbf{p}_{Gb_y} \\ \mathbf{p}_{Gb_z} \end{bmatrix}, \quad (5.44)$$

where  $\mathbf{p}_{Gb}$  is expressed in the inertial frame. This relative motion is used as inputs of the modified centroidal momentum/dynamics models, detailed in Section 5.3.6. In Section 5.3.3, we define  $\mathbf{P}_{Gb_x}$  and  $\mathbf{P}_{Gb_y}$  to be zero. It means that as long as either the base or the CoM translational motions is defined in the forward and lateral directions of the inertial frame, the other can be calculated with the same values.

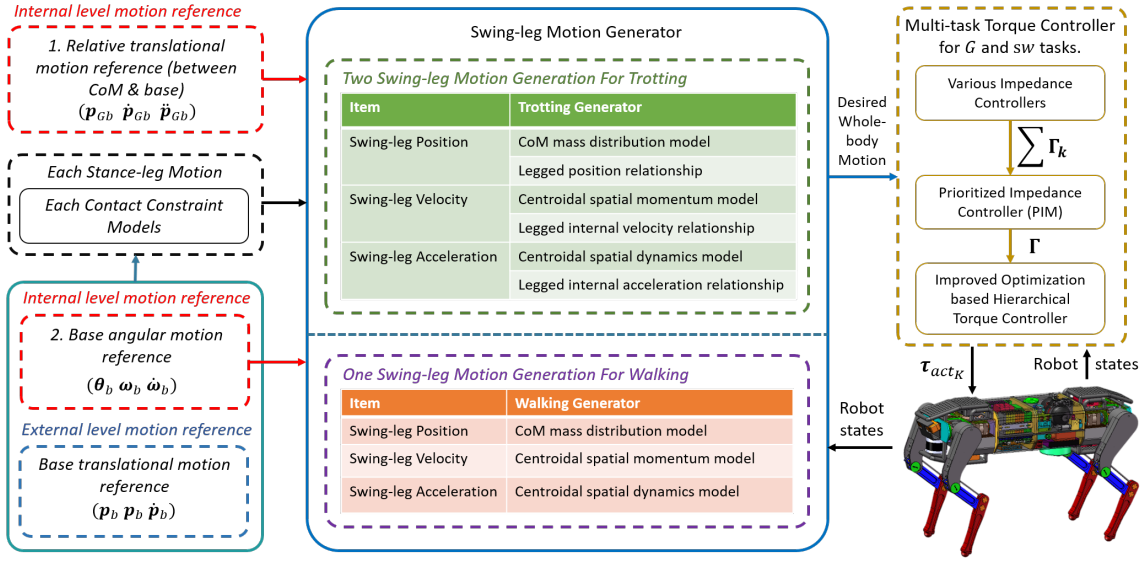


Figure 5.5: Whole-body Control Framework: The motion references include the external and internal motion levels. The internal motion references are used for the motion generation of walking and trotting gaits. In the motion generator, we achieve the swing-leg motion using the CoM mass distribution model, the centroidal momentum/dynamics models and the models relating to legged motion relationships. The two level motion references and the outputs of the motion generators are used as inputs of the torque controller. The variables in the framework are explained throughout this chapter.

The internal motion references are used for the motion generators of the walking and trotting modes, which output the swing-leg motion references. For the walking locomotion mode, only the mass distribution model, the centroidal translational momentum/dynamics models are applied. For the trotting locomotion mode, the extra centroidal angular momentum and dynamics models are applied, as well as the legged motion relationships among each leg.

For the torque controller, the motion references relating to body positions and velocities are transferred to the feedback based prioritized impedance controller. The motion references relating to body accelerations are used in the optimization based prioritized multi-task torque controller. The torque controller outputs the actuated torque to enable the robot to run various rough terrains.

In this way, we need to define the motions of the two levels. The internal properties enable to generate the adaptive legged motion to follow the external level motion automatically. It enables the robot to walk, trot, turn, etc. For the push-recovery motion, it influences the base translational motion in the external level and the base angular motion in the internal level, this will be handled by the compliance torque controller.

### 5.3.3 External Motion Reference

We define the base translational motion using several connected Bézier curves. The  $j$ 's Bézier curve relating to the base velocity can be written as follows,

$$\dot{\mathbf{p}}_{b|j} = \sum_{k=0}^{N_b} \frac{N_b!}{k!(N_b - k)!} \bar{t}^k (1 - \bar{t})^{N_b - k} \mathbf{P}_{k|j} \quad (5.45)$$

where  $\mathbf{P}_{k|j}$  denotes the  $k^{th}$  3D control point of the  $j$ 's Bézier curve. We define  $N_b = 4$  and there are five points for the  $j$ 's Bézier curve. We give one example in Fig. 5.6 relating to the base velocity in the

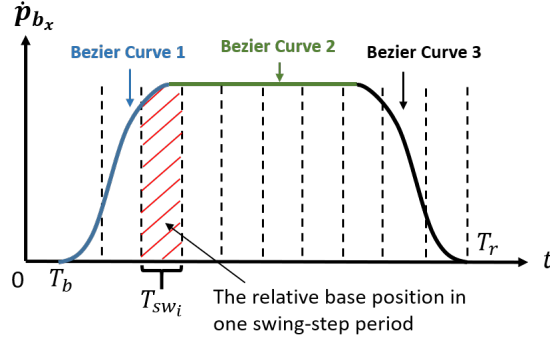


Figure 5.6: An example of the base forward motion reference with three Bézier curves, two curves for acceleration and deceleration, one linear curve for uniform velocity motion. The whole reference is divided into several equal phases. Each phase represents one swing-step period.

forward direction in the inertial frame. In addition,  $\bar{t} \in [0, 1]$  is the normalized time as below,

$$\bar{t} = \frac{t - t_{b|j}}{t_{e|j} - t_{b|j}}, \quad (5.46)$$

where  $t_{b|j}$  and  $t_{e|j}$  represent the start and ending time instances of the Bézier curve  $j$ . The integral and derivative of (5.45) denote the base position and acceleration respectively. The derivative with respect to time of one Bézier curve is also one Bézier curve. The integral of one Bézier curve can be referred in [Doha 2011]. In Fig. 5.6, we set  $\mathbf{P}_{0|j}$  and  $\mathbf{P}_{1|j}$  to be the same which enables the acceleration to be zero at the beginning of the  $j$ 's curve, as well as  $\mathbf{P}_{3|j}$  and  $\mathbf{P}_{4|j}$ . To enable a smooth transition from one curve to another, their positions, velocities and accelerations should be respectively equal at the connecting points. The curve can not only annotate the acceleration and deceleration motions, but also the uniform linear motion.

Given the base forward motion reference in Fig. 5.6, the whole reference is divided into  $N$  phases and each phase represents one swing-step period, shown in Fig. 5.6 where  $T_b$  and  $T_r$  are the beginning and ending time of the whole reference period, shown in Fig. 5.6. Maes et al. [Maes 2008] showed that the swing period of dogs' running experiments remains relatively constant over a wide range of speeds. In this state of the art, the constant desired swing phase period is always prescribed around 0.25s [Hyun 2014]. We also define  $i$ 's step period  $T_{sw_i} = 0.25s$  which is also the normal step period for all steps. In the real running situation, one step period can be larger or less than  $T_{sw_i}$ , detailed in next subsection. In this way, the beginning and end time instances for each step can be simply achieved. The relative base motion in one step period can be calculated using (5.45). For one step period consisting in two parts of two Bézier curves shown in the red part of Fig. 5.6, the relative base motion can also be calculated using two Bézier-curve functions (5.45). However, for the trotting mode without the four-leg flight phase, the constant step period enforces the robot to have big step distance at fast speeds which conflicts the leg-joint position limits. For this case, the quadruped animals will change the gait from the slow trotting mode to high running mode with the four-leg flight phase, which exceeds the topic of this chapter. We apply this segmentation method and gives the robot a high-level motion reference. Each step period is defined and ensure the sum of all the step periods to be equal to  $T_r - T_b$ . In addition, the step number  $N$  is adjusted to be an even integer which enables all legs to have the same step numbers. This work could be easily extended by applying a predictive control method and optimize the phase os each swing period and the step number by given one target and the whole-process time.

From the locomotion principle, the stance legs follow the base motion and then the swing-leg motion can be achieved. In the vertical direction, we select to control the relative motion between the CoM and the base ( $\mathbf{p}_{Gb_z}, \dot{\mathbf{p}}_{Gb_z}, \ddot{\mathbf{p}}_{Gb_z}$ ). For the legged generator in the forward and lateral directions, we simply define the centroidal motion to be similar with the floating-base motion. We give one example in the robot sagittal plane, shown in Fig. 5.7 in which the robot runs on one slope. The base and centroidal

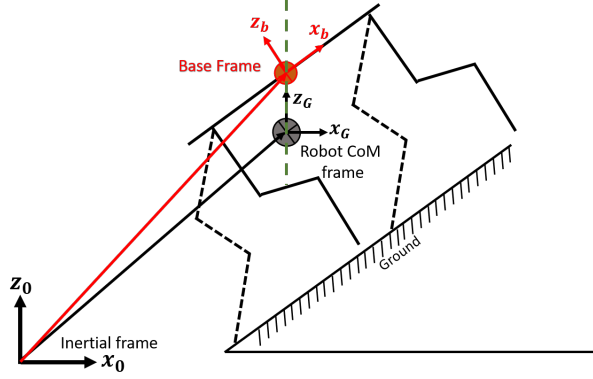


Figure 5.7: The motion relationship between the centroidal motion and the base motion on one slope. The dash line connecting the frame origins of the CoM and the base is set to be parallel with the  $z_0$  or  $z_G$  axis.

motions in the forward and lateral directions have the following relationships,

$$\mathbf{S}_{xy}\mathbf{p}_b = \mathbf{S}_{xy}\mathbf{p}_G, \quad (5.47a)$$

$$\mathbf{S}_{xy}\dot{\mathbf{p}}_b = \mathbf{S}_{xy}\dot{\mathbf{p}}_G, \quad (5.47b)$$

$$\mathbf{S}_{xy}\ddot{\mathbf{p}}_b = \mathbf{S}_{xy}\ddot{\mathbf{p}}_G, \quad (5.47c)$$

where  $\mathbf{S}_{xy} = [\mathbf{1}_{2 \times 2} \ \mathbf{0}_{2 \times 1}]$  selects the first and second components of any 3D vector. Then we have the following relationships by combining the relationships in (4.2) and (3.38) as below,

$$\mathbf{S}_{xy}\mathbf{v}_G = \mathbf{S}_{xy} {}^b\mathbf{R}_G^T \mathbf{v}_b, \quad (5.48a)$$

$$\mathbf{S}_{xy}\dot{\mathbf{v}}_G = \mathbf{S}_{xy} [{}^b\mathbf{R}_G^T (\dot{\mathbf{v}}_b + \boldsymbol{\omega}_b \times \mathbf{v}_b) - \boldsymbol{\omega}_G \times \mathbf{v}_G]. \quad (5.48b)$$

### 5.3.4 Internal Motion Reference

From the external motion reference, the desired relative motion between the centroidal motion and the base motion can be defined as follows,

$$\mathbf{p}_{Gb} = \mathbf{p}_G - \mathbf{p}_b = \begin{bmatrix} 0 \\ 0 \\ \mathbf{p}_{Gb_z} \end{bmatrix}. \quad (5.49)$$

In this subsection, we focus on the vertical centroidal motion reference, which determines the swing-leg motion. This reference should have a bump shape in one step, shown in Fig. 5.8. The variables in this subsection are described for the current step and the step index is omitted.

It is noticed that in the running process for the trotting mode, one swing leg may touch the ground first and then only one leg still swings during a short period of time. We will handle this case by the walking mode with three legs contacting with the ground. Therefore, the centroidal reference should ensure the two swing-legs to both touch the ground, even with different sequences. We propose to use the relative motion between the robot CoM and the base frames as the vertical reference (see Fig. 5.8). This relative motion reference in one step period is defined using two connected curves as follows,

$$\mathbf{p}_{Gb_z} = \sum_{k=0}^{N_a} \frac{N_a!}{k!(N_a - k)!} \bar{t}_a^k (1 - \bar{t}_a)^{N_a - k} A_k, \quad (5.50a)$$

$$\dot{\mathbf{p}}_{Gb_z} = \sum_{k=0}^{N_b} \frac{N_b!}{k!(N_b - k)!} \bar{t}_b^k (1 - \bar{t}_b)^{N_b - k} B_k, \quad (5.50b)$$



Table 5.2: Control Point Definition

$A_k$	Value	$B_k$	Value
$A_0$	$H$	$B_0$	$N_a(A_5 - A_4)$
$A_1$	$H$	$B_1$	$\frac{N_a(N_a-1)(A_5-2A_4+A_3)}{N_b} + B_0$
$A_2$	$H$	$B_2$	$\frac{5H-5A_5-(\frac{N_b}{B_0+B_1})}{3}$
$A_3$	$H + \frac{H_{Gb}-H}{1.4}$	$B_3$	$B_2$
$A_4$	$H_{Gb}$	$B_4$	$B_2$
$A_5$	$H_{Gb}$		

where  $N_a$  and  $N_b$  are defined to be 5 and 4 respectively.  $A_k$  and  $B_k$  represent the control points for the Bézier curves relating to  $\mathbf{p}_{Gb}$  and  $\dot{\mathbf{p}}_{Gb}$  respectively. For the first curve, the velocity and acceleration at the beginning point are zero, and the velocity at the end point should be zero. The two curves should be continuous at the connecting point. It is noticed that we define the second curve in the velocity level. Since two coincident control points lead to zero differential value, and three coincident control points result in zero double-differential value. In the end of the second curve, we prefer the acceleration to be zero, but the velocity should not be zero.  $\bar{t}_a$  and  $\bar{t}_b$  represent the normalized times in the range  $[0, 1]$  as follows,

$$\bar{t}_a = \frac{t - t_{o_i}}{t_{m_i} - t_{o_i}} = \frac{t - t_{o_i}}{0.5T_{sw_i}}, \quad (5.51a)$$

$$\bar{t}_b = \frac{t - t_{m_i}}{t_{d_i} - t_{m_i}} = \frac{t - t_{m_i}}{0.5T_{sw_i}}, \quad (5.51b)$$

where  $t_{o_i}$ ,  $t_{m_i}$  and  $t_{d_i}$  denote the lift-off, middle and touch-down timings for the  $i$ 's step. The first curve accounts for 50% of the step period. The definition of the control points satisfy:

- $A_0$ ,  $A_1$  and  $A_2$  are equal which ensures that the velocity and acceleration at  $t_{o_i}$  are zero.
- $A_4 = A_5$ , resulting that the velocity at  $t_{m_i}$  is zero.
- $B_0$ ,  $B_1$  and  $B_2$  (dependent on  $A_k$ ) are calculated to achieve the connecting continuity of two curves.
- $B_2$ ,  $B_3$  and  $B_4$  are equal, the acceleration and jerk in the end of the step period are zero, and the integral of the second arrives at  $H$  at  $\bar{t}_b$ .

The definition results of the control points are shown in Tab. 5.2, in which  $H_{Gb}$  is the curve amplitude.  $H$  is the normal value of  $\mathbf{p}_{Gb_z}$ .  $H_{d_i}$  represents the values at the end of the  $i$ 's step respectively.

We define that the two swing legs for the trotting gait should touch the ground at the same time after  $T_{sw_i}$  period. However, when the robot runs even on the flat terrain, the two swing legs may not land at the same time if the orientation angles of the floating base is not equal to the desired values. In this case, we still allow the curve reference to continue downward to adapt to the real-time states, shown in the first curve of Fig. 5.8, until the detection of the contact signals from two swing legs. In the next swing step achieved by the other two diagonal legs, the cure reference begins at the previous step-end value and enables the robot to recover to the normal state (see the second curve of Fig. 5.8). Therefore, the step period for the first step ( $t_{d_1} - t_{o_1}$ ) is larger than  $T_{sw_1}$ . If this case happens on the second step, our approach will be used again. Another case is that two legs both land before the step period as shown in Fig. 5.9. In the simulation Section, the first case in Fig. 5.8 always happens. For the first case, if the current step  $i$  is not finished after  $T_{sw_i}$ , the curve will still goes down by maintaining the velocity at  $t = t_{o_i} + T_{sw_i}$ , and the position value  $\mathbf{p}_{Gb_z}$  is expressed as follows,

$$\mathbf{p}_{Gb_z} = H + v_{Gb}(t - t_{o_i} - T_{sw_i}), \quad \text{if } t > t_{o_i} + T_{sw_i}, \quad (5.52)$$

where  $v_{Gb}$  is one defined velocity to enable the CoM to continue the landing-down motion.

It is noticed that, to track the robot centroidal nonzero-velocity motion, there is no stance phase for all four legs. Because two pairs of diagonal legs should move in symmetry configuration. The CoM

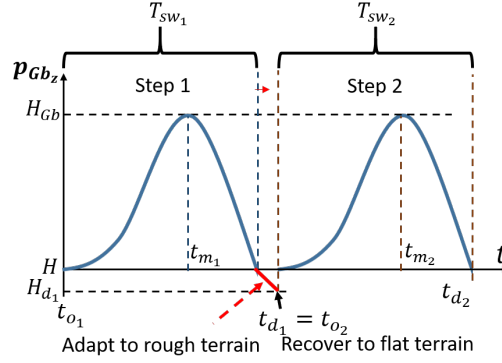


Figure 5.8: This shows two adjacent strides, consisting two swing-leg steps. In the first step period, the two swing legs do not land by following the  $p_{Gb_z}$  reference, another reference (with red color) is added to enable the robot to adapt to the real-time states. The second curve recovers to the normal state on the flat terrain.

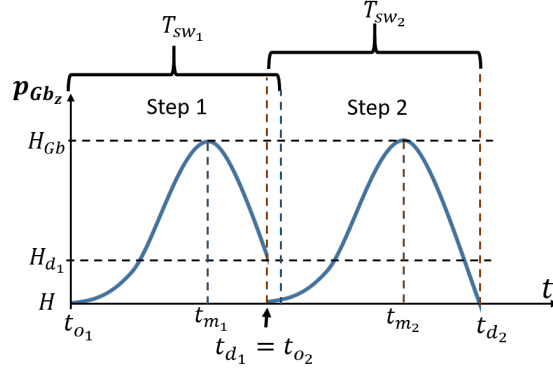


Figure 5.9: In the first step period, the step period is less than the defined value.

position of the 3-DOF stance leg is determined by the base motion. However, for humans and quadruped animals at slow speeds, all legs may touch the ground for short periods. We believe it is due to the fact that the extra sole DOF (one leg has more than 3 DOFs) enables the humans and quadruped animals to adjust their leg CoM position with bigger movement space without conflicting the contact situation.

The internal motion reference also includes the base angular motion, which serves as the inputs to generate the walking and trotting locomotion modes. If the vision is available for geometry estimation, the angular motion (orientation, velocity and acceleration) can be generated online.  $\theta_b$  can be pre-defined as several connected Bézier curves, and each Bézier curve can be expressed to be similar as in (5.45). In addition,  $\dot{\theta}_b$  and  $\ddot{\theta}_b$  can be achieved easily and they are also Bézier curves. The whole angular base motion can also be drawn similarly as Fig. 5.6, and it can also be divided by the two segmentation methods in Section 5.3.3. The desired angular base velocity  $\omega_b$  and acceleration  $\dot{\omega}_b$  can be derived by  $(\theta_b, \dot{\theta}_b, \ddot{\theta}_b)$  using (3.36). For the walking mode, the angular base motion references  $(\omega_b, \dot{\omega}_b)$  serve as the inputs of the translational centroidal models to derive the only-one swing-leg motion. For the trotting mode, the extra angular centroidal momentum and dynamics models are required and the base angular motion references  $(\omega_b, \dot{\omega}_b)$  also serve as the inputs, cf. Fig. 5.5.

For quadruped robots with no vision, the method in [Gehring 2015] can be applied to test the ground Euler angles online which can be used to enable the floating base to be parallel with the average ground geometry, however, the estimation method has delay by applying the contact situations of the previous three steps. In this chapter, we focus on the new locomotion principle and simply set the desired base roll and pitch orientation, angular velocity and acceleration around  $x_b$  and  $y_b$  axes to be zero on the flat

---

**Algorithm 2** State machine for switching the swing and stance legs of the continuous trotting

---

**Initialization:** step index  $i = 1$ ,  $t_{o_1} = T_b$

**if**  $t < T_b$  (When the robot stance with four legs) **then**

$\mathbf{S}_{control} = \mathbf{S}_{contact} = [1\ 1\ 1\ 1]$

$n_{level} = 1$

**else**

$n_{level} = 2$

**if**  $t \geq t_{o_i}$  (Start trotting with step  $i$ ) **then**

**if**  $i \% 2 = 1$  **then**

for the trotting mode with swing FL-HR legs:

$\mathbf{S}_{control} = [0\ 1\ 0\ 1]$

**end**

**if**  $i \% 2 = 0$  **then**

for the trotting mode with swing FR-HL legs:

$\mathbf{S}_{control} = [1\ 0\ 1\ 0]$

**end**

**if**  $\sum \mathbf{S}_{contact} = 3$  and  $(t - t_{o_i}) \geq 0.6T_{sw_i}$  **then**

for the walking mode with only one swing leg:

$\mathbf{S}_{control} = \mathbf{S}_{contact}$

**end**

**if**  $[\sum \mathbf{S}_{contact} = 4$  and  $(t - t_{o_i}) \geq 0.6T_{sw_i}]$  or  $[(t - t_{o_i}) \geq \sigma T_{sw_i}]$  **then**

for switching the trotting leg indexes:

$t_{d_i} = t$

$t_{o_{i+1}} = t$

next step index  $i = i + 1$

$\mathbf{S}_{control} = [0\ 1\ 0\ 1]$ , if  $i \% 2 = 1$

$\mathbf{S}_{control} = [1\ 0\ 1\ 0]$ , if  $i \% 2 = 0$

**end**

**end**

**end**

---

terrain, namely,  $\alpha = \beta = 0$  and  $\omega_{b_x} = \omega_{b_y} = 0$ . However, the angular motion around  $z_b$  is not zero when the robot turns.

### 5.3.5 State Machine

In the whole-body control framework for trotting, c.f. Fig. 5.5, the walking mode sometimes happens, depending on whether both swing-legs touch the ground at the same time or not. Therefore, in this subsection, we develop one state machine to generate the trotting mode, shown in Algorithm 2, which enables to achieve the swing or stance leg indexes. We define two variable  $\mathbf{S}_{control}$  and  $\mathbf{S}_{contact}$  to represent the control and feedback signals of the contact situation as,

$$\mathbf{S}_{control} = [1/0\ 1/0\ 1/0\ 1/0], \quad (5.53a)$$

$$\mathbf{S}_{contact} = [1/0\ 1/0\ 1/0\ 1/0], \quad (5.53b)$$

where the two signals both contain four components corresponding to the control/contact signals in  $i$ 's step for the legs FL, FR, HR, HL in sequence, and each component is equal to 1 or 0. For  $\mathbf{S}_{control}$ , 1 denote that the corresponding leg is controlled to be a stance leg, and 0 means that the leg is activated to be a swing leg.  $\mathbf{S}_{contact}$  is one feedback signal variable, 1 means that the contact sensor gives back that the leg touches the ground, and 0 represents that the leg has no contact with the ground. It is noticed that the two variables may not be the same. For example, at the beginning the one swing step, all legs are still in contact with the ground,  $\mathbf{S}_{contact} = [1\ 1\ 1\ 1]$ , however,  $\mathbf{S}_{control}$  only has two 1 inside. In addition,  $\mathbf{S}_{control}$  also depends on  $\mathbf{S}_{contact}$ , e.g. when the robot trots on the uneven terrain, one swing leg touches the ground first. In this case,  $\mathbf{S}_{contact}$  has three 1 inside, then  $\mathbf{S}_{control}$  should be equal to  $\mathbf{S}_{contact}$  until the second swing leg touching the ground. After the two swing legs both touch the ground,

$\mathbf{S}_{control}$  and  $\mathbf{S}_{contact}$  change together. In Algorithm 2, we define the relationship detailedly,  $\sum \mathbf{S}_{contact}$  equals to the sum of the items in  $\mathbf{S}_{contact}$ , and the algorithm is executed in each control loop. For the switching algorithm respecting  $\sum \mathbf{S}_{contact} = 4$ , this part is executed only when the swing legs experience more than half (60%) of the normal defined swing period. We set that the switching configuration also happens when the swing legs experience  $\sigma$  times of the normal step period and in the simulation section,  $\sigma = 1.3$  for the push-recovery motion,  $\sigma = 1.6$  for trotting on rough terrains and  $\sigma = 1.2$  for other scenarios. It is noticed that the size of the contact Jacobian depends on  $\mathbf{S}_{control}$ , not  $\mathbf{S}_{contact}$ . In this algorithm, we also output the control hierarchy number.  $n_{level} = 1$  means that we only control one hierarchical task (centroidal motion tracking task).  $n_{level} = 2$  denotes that two hierarchical tasks are controlled, including the centroidal motion tracking task and one swing-leg tracking task. No matter how many hierarchies are controlled, they should respect the stance-leg contact constraints. The control hierarchy is detailed in Section 5.4.

### 5.3.6 Modified Centroidal Models For Swing-leg Motion Generators

Initially, we use the relative motion between the base and the CoM  $\mathbf{p}_{Gb}$  to rewrite the centroidal translational models using (5.47)-(5.48) as follows,

$$\dot{\mathbf{p}}_{Gb} = \frac{m}{m_G} {}^b \mathbf{R}_G^T \left[ \sum_{i=1}^4 (\boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + \dot{\tilde{\mathbf{p}}}_i) \right] = \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ \dot{\tilde{\mathbf{p}}}_{Gb_z} \end{bmatrix}, \quad (5.54a)$$

$$= {}^b \mathbf{R}_G^T \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_G + \frac{m}{m_G} {}^b \mathbf{R}_G^T \sum_{i=1}^4 \dot{\tilde{\mathbf{p}}}_i, \quad (5.54b)$$

and

$$\ddot{\mathbf{p}}_{Gb} = \frac{m}{m_G} {}^b \mathbf{R}_G^T \left[ \mathbf{s} + \sum_{i=1}^4 (\dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_i + \ddot{\tilde{\mathbf{p}}}_i) \right] = \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ \ddot{\tilde{\mathbf{p}}}_{Gb_z} \end{bmatrix}, \quad (5.55a)$$

$$= {}^b \mathbf{R}_G^T \dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_G + \frac{m}{m_G} {}^b \mathbf{R}_G^T \mathbf{s} + \frac{m}{m_G} {}^b \mathbf{R}_G^T \sum_{i=1}^4 \ddot{\tilde{\mathbf{p}}}_i, \quad (5.55b)$$

where  $\mathbf{s}$  is derived as below,

$$\mathbf{s} = \sum_{i=1}^4 [\boldsymbol{\omega}_b \times (\boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i + 2\dot{\tilde{\mathbf{p}}}_i)]. \quad (5.56)$$

The angular inertia at the robot CoM  $\bar{\mathbf{I}}_G$  is derived as below,

$$\bar{\mathbf{I}}_G = {}^b \mathbf{R}_G^T \bar{\mathbf{I}}_b {}^b \mathbf{R}_G - {}^b \mathbf{R}_G^T \tilde{\mathbf{p}}_G \times m_b \tilde{\mathbf{p}}_G \times {}^b \mathbf{R}_G + \sum_{i=1}^4 ({}^b \mathbf{R}_G^T \bar{\mathbf{I}}_i {}^b \mathbf{R}_G - {}^b \mathbf{R}_G^T {}^i \mathbf{p}_G \times m^i \mathbf{p}_G \times {}^b \mathbf{R}_G), \quad (5.57)$$

therefore, we can derive the following relationship by substituting (5.31b) and (5.38) into (5.57),

$${}^b \mathbf{R}_G \bar{\mathbf{I}}_G {}^b \mathbf{R}_G^T \boldsymbol{\omega}_b = \bar{\mathbf{I}}_b \boldsymbol{\omega}_b + \sum_{i=1}^4 [\bar{\mathbf{I}}_i \boldsymbol{\omega}_b - {}^i \mathbf{p}_G \times m \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_i]. \quad (5.58)$$

From the centroidal angular momentum model in (5.39a) and the above relationship, we can extract the following relationship as follows,

$${}^b \mathbf{R}_G \bar{\mathbf{I}}_G \boldsymbol{\omega}_G = \sum_{i=1}^4 (\bar{\mathbf{I}}_i \mathbf{L}_i \dot{\tilde{\mathbf{p}}}_i - {}^i \mathbf{p}_G \times m \dot{\tilde{\mathbf{p}}}_i) + {}^b \mathbf{R}_G \bar{\mathbf{I}}_G {}^b \mathbf{R}_G^T \boldsymbol{\omega}_b. \quad (5.59)$$

Similarly, we can achieve the legged CoM acceleration relationship using the centroidal angular momentum model in (5.42a) as,

$${}^b \mathbf{R}_G \bar{\mathbf{I}}_G \dot{\boldsymbol{\omega}}_G = \sum_{i=1}^4 (\bar{\mathbf{I}}_i \mathbf{L}_i \ddot{\tilde{\mathbf{p}}}_i - {}^i \mathbf{p}_G \times m \ddot{\tilde{\mathbf{p}}}_i) + {}^b \mathbf{R}_G \bar{\mathbf{I}}_G {}^b \mathbf{R}_G^T \dot{\boldsymbol{\omega}}_b - \mathbf{w}, \quad (5.60)$$

where  $\mathbf{w}$  depends on  $\mathbf{e}$  in (5.43a) as follows,

$$\mathbf{w} = {}^b\mathbf{R}_G (\boldsymbol{\omega}_G \times \bar{\mathbf{I}}_G \boldsymbol{\omega}_G) - \mathbf{e} - \sum_{i=1}^4 \bar{\mathbf{I}}_i (\boldsymbol{\omega}_b \times \tilde{\boldsymbol{\omega}}_i - \mathbf{Q}_i). \quad (5.61)$$

It is noticed that the models in this subsection are independent of the translational components of the base and the centroidal motions, and the legged motion can be achieved on-line only using the robot internal properties.

### 5.3.7 Desired Composite Swing-leg Motion

The legged motion generator is built based on our proposed locomotion principle using each leg centroidal model and the whole-robot centroidal model in Section 5.2. For gaits with  $\geq 2$  contact points, we divide the four legs into two groups, one group consisting of the stance legs (symbolically as  $st$ ) and one group for the swing legs (symbolically as  $sw$ ). The stance legs follow the base motion and their desired motions are derived in Section 5.2.2.3. For the walking locomotion mode, there are 3 stance legs and 1 swing leg. For the trotting locomotion mode, there are 2 stance legs and 2 swing legs. We will derive the desired composite swing-leg motion in this section, and then achieve each swing-leg motion in the following sections. The desired swing-leg-group motion can be achieved based on the desired stance-leg-group motion. Initially, the desired swing-leg positions can be derived using the mass distribution model in (5.31) as below,

$$\sum_{i=1}^{n_{sw}} \tilde{\mathbf{p}}_{sw_i} = \underbrace{{}^b\mathbf{R}_G \frac{m_G}{m} \mathbf{p}_{Gb}}_{\Pi_1} - \sum_{j=1}^{n_{st}} \tilde{\mathbf{p}}_{st_j}, \quad (5.62)$$

where  $\tilde{\mathbf{p}}_{st_j}$  and  $\tilde{\mathbf{p}}_{sw_i}$  represent the CoM positions of the stance leg  $j$  and the swing leg  $i$  in the base frame respectively. For the walking and trotting locomotion modes,  $n_{sw}$  equals to 1 and 2 respectively,  $n_{st}$  is equal to 3 and 2 respectively. The desired swing-leg velocities can be achieved based on the modified centroidal momentum model in (5.54) and (5.59) as,

$$\sum_{i=1}^{n_{sw}} \dot{\tilde{\mathbf{p}}}_{sw_i} = \underbrace{\frac{m_G}{m} {}^b\mathbf{R}_G \dot{\mathbf{p}}_{Gb} - \frac{m_G}{m} \boldsymbol{\omega}_b \times \tilde{\mathbf{p}}_G}_{\Pi_2} - \sum_{j=1}^{n_{st}} \dot{\tilde{\mathbf{p}}}_{st_j}, \quad (5.63a)$$

$$\sum_{i=1}^{n_{sw}} \underbrace{(\bar{\mathbf{I}}_{sw_i} \mathbf{L}_{sw_i} - {}^{sw_i}\mathbf{p}_G \times)}_{\Omega_{sw_i}} \dot{\tilde{\mathbf{p}}}_{sw_i} = {}^b\mathbf{R}_G \bar{\mathbf{I}}_G \boldsymbol{\omega}_G - \Xi_1, \quad (5.63b)$$

where  ${}^{sw_i}\mathbf{p}_G$  and  $\bar{\mathbf{I}}_{sw_i}$  denote the robot centroidal position and angular inertia of the stance-leg  $i$  CoM.  $\mathbf{L}_{sw_i}$  is the coefficient matrix relating the swing leg  $i$ , defined in (5.29).  $\Xi_1$  is achieved as below,

$$\Xi_1 = {}^b\mathbf{R}_G \bar{\mathbf{I}}_G {}^b\mathbf{R}_G^T \boldsymbol{\omega}_b + \sum_{j=1}^{n_{st}} \underbrace{(\bar{\mathbf{I}}_{st_j} \mathbf{L}_{st_j} - {}^{st_j}\mathbf{p}_G \times)}_{\Omega_{st_j}} \dot{\tilde{\mathbf{p}}}_{st_j}, \quad (5.64)$$

where  ${}^{st_j}\mathbf{p}_G$  and  $\bar{\mathbf{I}}_{st_j}$  denote the robot centroidal position and angular inertia of the stance-leg  $j$  CoM.  $\mathbf{L}_{st_j}$  is the coefficient matrix relating the stance leg  $j$ , c.f. (5.29).

Similarly, the desired swing-leg accelerations can be achieved based on the modified centroidal dynamics model in (5.55) and (5.60) as follows,

$$\sum_{i=1}^{n_{sw}} \ddot{\tilde{\mathbf{p}}}_{sw_i} = \underbrace{\frac{m_G}{m} {}^b\mathbf{R}_G \ddot{\mathbf{p}}_{Gb} - \mathbf{s} - \frac{m_G}{m} \dot{\boldsymbol{\omega}}_b \times \tilde{\mathbf{p}}_G}_{\Pi_3} - \sum_{j=1}^{n_{st}} \ddot{\tilde{\mathbf{p}}}_{st_j}, \quad (5.65a)$$

$$\sum_{i=1}^{n_{sw}} \Omega_{sw_i} \ddot{\tilde{\mathbf{p}}}_{sw_i} = {}^b\mathbf{R}_G \bar{\mathbf{I}}_G \dot{\boldsymbol{\omega}}_G - \Xi_2, \quad (5.65b)$$

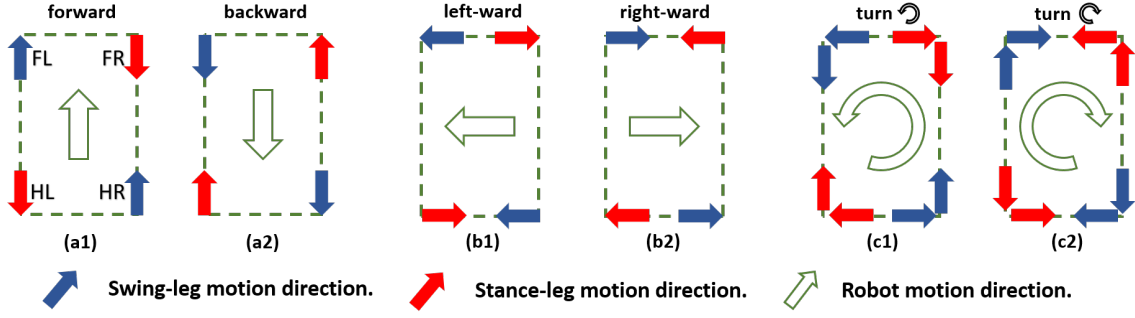


Figure 5.10: This shows the leg motion relationship for the trotting mode in the forward-backward direction as shown in (a1) and (a2), in the lateral direction as depicted in (b1) and (b2), and in the turning-in-place situations as presented in (c1) and (c2). We take the front-left (FL) and hind-right (HR) legs as the swing legs. The front-right (FR) and hind-left (HL) legs serve as the stance legs. The blue and red arrows denote the motion direction of the swing legs and stance legs respectively relative to the base. The green arrows represent the robot motion direction.

where  $\Xi_2$  is achieved as below,

$$\Xi_2 = {}^b R_G \bar{\mathbf{I}}_G^b R_G^T \dot{\omega}_b - \mathbf{w} + \sum_{j=1}^{n_{st}} \Omega_{stj} \ddot{\mathbf{p}}_{stj}. \quad (5.66)$$

In this subsection, we derive the desired swing-leg group motion based on the relative motion reference between the base and the CoM motions, and the desired stance-leg motion. Then we will derive each swing-leg motion in the following sections.

### 5.3.8 Desired Swing-leg Motion for Walking

For the walking locomotion mode with only one swing leg, the swing-leg motion generator applies the relative translational motion  $\mathbf{p}_{Gb}$  between the CoM and base motions as the motion reference. Therefore, only the modified translational centroidal models in (5.54) and (5.55) are applied. By given the desired stance-leg motions, the only-one swing-leg desired position  $\tilde{\mathbf{p}}_{sw}$  can be easily derived, as well as the velocity and acceleration,

$$\tilde{\mathbf{p}}_{sw} = (5.62)right, \quad (5.67a)$$

$$\dot{\tilde{\mathbf{p}}}_{sw} = (5.63a)right, \quad (5.67b)$$

$$\ddot{\tilde{\mathbf{p}}}_{sw} = (5.65a)right, \quad (5.67c)$$

where *right* denotes the right side of one equation. *sw* denotes the swing-leg index for walking with only one swing leg. It is noticed that the desired swing-leg position is achieved using the actual stance-leg position. The desired swing-leg CoM velocity is derived using the actual robot configuration, not the desired values in (5.67a). In addition, the desired swing-leg CoM acceleration is derived using the actual robot configuration and velocity, not the desired values in (5.67a) and (5.67b). Then we can achieve the desired centroidal angular velocity  $\omega_G$  and acceleration  $\dot{\omega}_G$  using (5.59) and (5.60) respectively. The desired position and velocity references are used in the impedance controller, and the desired acceleration references are applied in the feed-forward torque controller. In this chapter, although we concentrate on the trotting mode, the walking mode is applied when the two trotting swing legs land on the ground at different timings. In the simulations, for almost trotting steps, this walking mode always happens for short periods. In addition, the derived equations in (5.67) can also be used in the trotting locomotion gait by setting  $\tilde{\mathbf{p}}_{sw} = \tilde{\mathbf{p}}_{sw_f} + \tilde{\mathbf{p}}_{sw_h}$ , in which  $\tilde{\mathbf{p}}_{sw_f}$  and  $\tilde{\mathbf{p}}_{sw_h}$  represent the front and hind swing legs respectively, and *sw<sub>f</sub>* and *sw<sub>h</sub>* denote the swing-leg indexes.

### 5.3.9 Leg-Motion Relationship for Trotting

In this chapter, the trotting gait means that the two-swing diagonal legs move together. Compared with conventional trotting gaits in the state of the art, we suppose that the two swing-legs can have different motion. We show three situations for one step, shown in Fig. 5.10, in which the FL-HR legs serve as the swing legs and the motion directions of the four legs are depicted relative to the base. We depict situations when the robot runs in the forward and lateral directions in Fig. 5.10(a) and Fig. 5.10(b), and turns in place in Fig. 5.10(c). Each situation contains two sub-situations in opposite directions. It is noticed that for the three situations in Fig. 5.10, the motion direction of the swing-FL leg is always opposite to that of the stance-HL leg in the forward direction, and it is always opposite to that of the stance-FR leg in the lateral direction. Similarly, the motion direction of the swing-HR leg is always opposite to those of the stance FR and HL legs in the forward and lateral directions respectively. Even though the situations in Fig. 5.10 can represent the directions of the relative positions and velocities of four legs relative to the base, the motion directions relating to the relative acceleration directions also respect to this principle. In addition, we assume that the absolute values for the four legged motions are the same in the vertical directions. Therefore, in this subsection, we define the swing-leg motion by combining the centroidal position distribution in (5.62), the centroidal translational momentum and dynamics models in (5.63a) and (5.65a) respectively.

For the swing-leg relationships relating to the position, velocity and acceleration for the symmetric trotting in the base forward and lateral directions, we can define the following relationships by combining (5.62), (5.63a), (5.65a) and the internal relationships relating to the four-legged motion directions in Fig. 5.10,

$$\mathbf{S}_{xy}\tilde{\mathbf{p}}_{sw_f} = \mathbf{r}_{sw}\mathbf{S}_{xy}\mathbf{\Pi}_1 - \begin{bmatrix} \mathbf{S}_x\tilde{\mathbf{p}}_{st_h} \\ \mathbf{S}_y\tilde{\mathbf{p}}_{st_f} \end{bmatrix}, \quad (5.68a)$$

$$\mathbf{S}_{xy}\dot{\tilde{\mathbf{p}}}_{sw_f} = \mathbf{r}_{sw}\mathbf{S}_{xy}\mathbf{\Pi}_2 - \begin{bmatrix} \mathbf{S}_x\dot{\tilde{\mathbf{p}}}_{st_h} \\ \mathbf{S}_y\dot{\tilde{\mathbf{p}}}_{st_f} \end{bmatrix}, \quad (5.68b)$$

$$\mathbf{S}_{xy}\ddot{\tilde{\mathbf{p}}}_{sw_f} = \mathbf{r}_{sw}\mathbf{S}_{xy}\mathbf{\Pi}_3 - \begin{bmatrix} \mathbf{S}_x\ddot{\tilde{\mathbf{p}}}_{st_h} \\ \mathbf{S}_y\ddot{\tilde{\mathbf{p}}}_{st_f} \end{bmatrix}, \quad (5.68c)$$

where  $\mathbf{r}_{sw} = \text{diag}[0.5, 0.5]$  is one defined ratio matrix.  $\mathbf{S}_x = [1 \ 0 \ 0]$  and  $\mathbf{S}_y = [0 \ 1 \ 0]$ .  $\tilde{\mathbf{p}}_{st_f}$  and  $\tilde{\mathbf{p}}_{st_h}$  represent the front/hind stance-leg CoM positions respectively in the base frame, and  $st_f$  and  $st_h$  denote the stance-leg indexes for the trotting mode. It is noticed that the two stance-leg motion properties in two different directions are applied which correspond to the principle in Fig. 5.10. Then the swing-hind leg position, velocity and acceleration in the base forward and lateral directions can be derived by (5.62) – (5.68a), (5.63a) – (5.68b), (5.65a) – (5.68c) respectively as below,

$$\mathbf{S}_{xy}\tilde{\mathbf{p}}_{sw_h} = (\mathbf{1}_{2 \times 2} - \mathbf{r}_{sw})\mathbf{S}_{xy}\mathbf{\Pi}_1 - \begin{bmatrix} \mathbf{S}_x\tilde{\mathbf{p}}_{st_f} \\ \mathbf{S}_y\tilde{\mathbf{p}}_{st_h} \end{bmatrix}, \quad (5.69a)$$

$$\mathbf{S}_{xy}\dot{\tilde{\mathbf{p}}}_{sw_h} = (\mathbf{1}_{2 \times 2} - \mathbf{r}_{sw})\mathbf{S}_{xy}\mathbf{\Pi}_2 - \begin{bmatrix} \mathbf{S}_x\dot{\tilde{\mathbf{p}}}_{st_f} \\ \mathbf{S}_y\dot{\tilde{\mathbf{p}}}_{st_h} \end{bmatrix}, \quad (5.69b)$$

$$\mathbf{S}_{xy}\ddot{\tilde{\mathbf{p}}}_{sw_h} = (\mathbf{1}_{2 \times 2} - \mathbf{r}_{sw})\mathbf{S}_{xy}\mathbf{\Pi}_3 - \begin{bmatrix} \mathbf{S}_x\ddot{\tilde{\mathbf{p}}}_{st_f} \\ \mathbf{S}_y\ddot{\tilde{\mathbf{p}}}_{st_h} \end{bmatrix}. \quad (5.69c)$$

(5.68) and (5.69) mean that the two swing legs move in different forward/lateral directions, as well as the two stance two legs. Compared with the definition of the internal legged relationships in the state of the art that the swing-leg motion references have the opposite values with stance legs, our method can also handle the turning cases. In addition, another advantage of (5.68) and (5.69) is that they enable the legged acceleration references to be smooth. Therefore, two of [(5.62), (5.63a), (5.65a)], [(5.68a), (5.68b), (5.68c)] and [(5.69a), (5.69b), (5.69c)] are independent and will be used in the swing-leg motion generation in the next subsection.



The relationship relating to the vertical motion of the two swing legs are defined as follows,

$$\mathbf{S}_z^b \mathbf{R}_G^T (\tilde{\mathbf{p}}_{sw_f} - \mathbf{B}_{sw_f}) = \xi \mathbf{S}_z^b \mathbf{R}_G^T (\tilde{\mathbf{p}}_{sw_h} - \mathbf{B}_{sw_h}), \quad (5.70a)$$

$$\mathbf{S}_z^b \mathbf{R}_G^T \dot{\tilde{\mathbf{p}}}_{sw_f} = \xi \mathbf{S}_z^b \mathbf{R}_G^T \dot{\tilde{\mathbf{p}}}_{sw_h}, \quad (5.70b)$$

$$\mathbf{S}_z^b \mathbf{R}_G^T \ddot{\tilde{\mathbf{p}}}_{sw_f} = \xi \mathbf{S}_z^b \mathbf{R}_G^T \ddot{\tilde{\mathbf{p}}}_{sw_h}, \quad (5.70c)$$

where  $\mathbf{S}_z = [0 \ 0 \ 1]$  selects the third component of any 3D vector.  $\xi = 1$  means that the two swing legs share the similar motions.  $\mathbf{B}_{sw_f}$  and  $\mathbf{B}_{sw_h}$  are defined as follows,

$$\begin{aligned} \mathbf{B}_{sw_f} &= \begin{bmatrix} \frac{l_r}{2} & \frac{w_r}{2} & h_r \end{bmatrix}^T, & \text{for FL leg,} \\ &= \begin{bmatrix} \frac{l_r}{2} & -\frac{w_r}{2} & h_r \end{bmatrix}^T, & \text{for FR leg,} \end{aligned} \quad (5.71a)$$

$$\begin{aligned} \mathbf{B}_{sw_h} &= \begin{bmatrix} -\frac{l_r}{2} & \frac{w_r}{2} & h_r \end{bmatrix}^T, & \text{for HL leg,} \\ &= -\begin{bmatrix} \frac{l_r}{2} & \frac{w_r}{2} & h_r \end{bmatrix}^T, & \text{for HR leg,} \end{aligned} \quad (5.71b)$$

where  $h_r$  denotes the relative position between the base frame origin and each HAA joint origin, in  $z_b$  direction.  $l_r$  and  $w_r$  are referred in Fig. 5.2. For the normal terrains, the vertical motions between the swing legs are set to be equal,  $\xi = 1$ . However, when the robot climbs stairs in artificial buildings,  $\xi$  can be set to be different to enable one swing leg lifts higher than the other swing leg. (5.70) will be used in the swing-leg motion generation in the next subsection.

### 5.3.10 Desired Swing-leg Motions for Trotting

The type of the motion references for the trotting gait is the same with the walking gait. For the swing-leg motion generation, we will add the internal relationships among legs. Initially, the desired positions of the two swing legs share the following relationships,

$$\tilde{\mathbf{p}}_{sw_f} + \tilde{\mathbf{p}}_{sw_h} = (5.62)right, \quad (5.72a)$$

$$\begin{bmatrix} \mathbf{S}_{xy} \tilde{\mathbf{p}}_{sw_f} \\ \mathbf{S}_z^b \mathbf{R}_G^T (\tilde{\mathbf{p}}_{sw_f} - \xi \tilde{\mathbf{p}}_{sw_h}) \end{bmatrix} = \begin{bmatrix} (5.68a)right \\ \mathbf{S}_z^b \mathbf{R}_G^T (\mathbf{B}_{sw_f} - \xi \mathbf{B}_{sw_h}) \end{bmatrix}, \quad (5.72b)$$

where (5.72a) is derived using the centroidal position relationship in (5.62). (5.72b) is the combination of (5.68a) and (5.70a). (5.72) ensures that the swing legs experience the same distance with the delay stance legs in the base forward and lateral directions. Then the desired two swing-leg position can be derived as follows,

$$\underbrace{\begin{bmatrix} \mathbf{1}_{3 \times 3} & \mathbf{1}_{3 \times 3} \\ \mathbf{S}_{xy} & \mathbf{0}_{2 \times 3} \\ \mathbf{S}_z^b \mathbf{R}_G^T & -\xi \mathbf{S}_z^b \mathbf{R}_G^T \end{bmatrix}}_{\mathbf{A}_p \in \mathbb{R}^{6 \times 6}} \begin{bmatrix} \tilde{\mathbf{p}}_{sw_f} \\ \tilde{\mathbf{p}}_{sw_h} \end{bmatrix} = \underbrace{\begin{bmatrix} (5.72a)right \\ (5.72b)right \end{bmatrix}}_{\mathbf{B}_p \in \mathbb{R}^{6 \times 1}}, \quad (5.73)$$

where the desired swing-leg positions  $(\tilde{\mathbf{p}}_{sw_f}, \tilde{\mathbf{p}}_{sw_h})$  can be achieved by  $\mathbf{A}_p^{-1} \mathbf{B}_p$ .

Similarly to the motion generator for the walking gait, the desired swing-leg CoM velocities for the trotting mode are derived using the actual robot configuration, not the desired values in (5.72). They can be expressed as follows,

$$\dot{\tilde{\mathbf{p}}}_{sw_f} + \dot{\tilde{\mathbf{p}}}_{sw_h} = (5.63a)right, \quad (5.74a)$$

$${}^b \mathbf{R}_G \bar{\mathbf{I}}_G \boldsymbol{\omega}_G - \boldsymbol{\Omega}_{sw_f} \times \dot{\tilde{\mathbf{p}}}_{sw_f} - \boldsymbol{\Omega}_{sw_h} \times \dot{\tilde{\mathbf{p}}}_{sw_h} = \boldsymbol{\Xi}_1, \quad (5.74b)$$

$$\begin{bmatrix} \mathbf{S}_{xy} \dot{\tilde{\mathbf{p}}}_{sw_f} \\ \mathbf{S}_z^b \mathbf{R}_G^T (\dot{\tilde{\mathbf{p}}}_{sw_f} - \xi \dot{\tilde{\mathbf{p}}}_{sw_h}) \end{bmatrix} = \begin{bmatrix} (5.68b)right \\ 0 \end{bmatrix}, \quad (5.74c)$$

where  $n_{sw} = n_{st} = 2$  for the trotting mode. (5.74a) is extracted from the modified translational centroidal momentum model. (5.74b) is equivalent to the centroidal angular momentum model in (5.59). (5.74c)

denote the internal relationships among the legs in the velocity level. Then the three equations in (5.74) are applied to achieve the desired swing-leg velocities and the centroidal angular velocity as,

$$\underbrace{\begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} & \mathbf{1}_{3 \times 3} \\ {}^b\mathbf{R}_G \bar{\mathbf{I}}_G & -\boldsymbol{\Omega}_{sw_f} & -\boldsymbol{\Omega}_{sw_h} \\ \mathbf{0}_{2 \times 3} & \mathbf{S}_{xy} & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{1 \times 3} & \mathbf{S}_z {}^b\mathbf{R}_G^T & -\xi \mathbf{S}_z {}^b\mathbf{R}_G^T \end{bmatrix}}_{\mathbf{A}_v \in \mathbb{R}^{9 \times 9}} \begin{bmatrix} \boldsymbol{\omega}_G \\ \dot{\tilde{\mathbf{p}}}_{sw_f} \\ \dot{\tilde{\mathbf{p}}}_{sw_h} \end{bmatrix} = \underbrace{\begin{bmatrix} (5.74a)right \\ (5.74b)right \\ (5.74c)right \end{bmatrix}}_{\mathbf{B}_v \in \mathbb{R}^{9 \times 1}}, \quad (5.75)$$

In this way, the desired  $[\boldsymbol{\omega}_G^T \ \dot{\tilde{\mathbf{p}}}_{sw_f}^T \ \dot{\tilde{\mathbf{p}}}_{sw_h}^T]^T = \mathbf{A}_v^{-1} \mathbf{B}_v$ .

Finally, we achieve the desired two swing-leg CoM accelerations and the desired centroidal angular acceleration for the trotting mode using the actual robot configuration and velocities as below,

$$\ddot{\tilde{\mathbf{p}}}_{sw_f} + \ddot{\tilde{\mathbf{p}}}_{sw_h} = (5.65a)right, \quad (5.76a)$$

$${}^b\mathbf{R}_G \bar{\mathbf{I}}_G \dot{\boldsymbol{\omega}}_G - \boldsymbol{\Omega}_{sw_f} \times \ddot{\tilde{\mathbf{p}}}_{sw_f} - \boldsymbol{\Omega}_{sw_h} \times \ddot{\tilde{\mathbf{p}}}_{sw_h} = \boldsymbol{\Xi}_2, \quad (5.76b)$$

$$\begin{bmatrix} \mathbf{S}_{xy} \ddot{\tilde{\mathbf{p}}}_{sw_f} \\ \mathbf{S}_z {}^b\mathbf{R}_G^T (\ddot{\tilde{\mathbf{p}}}_{sw_f} - \xi \ddot{\tilde{\mathbf{p}}}_{sw_h}) \end{bmatrix} = \begin{bmatrix} (5.68c)right \\ 0 \end{bmatrix}, \quad (5.76c)$$

where (5.76a) is extracted from the modified translational centroidal dynamics model. (5.76b) is equivalent to the centroidal angular dynamics model in (5.60). (5.76c) denotes the internal relationships among the legs in the acceleration level. Then the three equations in (5.76) are applied to achieve the desired  $[\dot{\boldsymbol{\omega}}_G, \ddot{\tilde{\mathbf{p}}}_{sw_f}, \ddot{\tilde{\mathbf{p}}}_{sw_h}]$  as,

$$\mathbf{A}_a \begin{bmatrix} \dot{\boldsymbol{\omega}}_G \\ \ddot{\tilde{\mathbf{p}}}_{sw_f} \\ \ddot{\tilde{\mathbf{p}}}_{sw_h} \end{bmatrix} = \begin{bmatrix} (5.76a)right \\ (5.76b)right \\ (5.76c)right \end{bmatrix} = \mathbf{B}_a, \quad (5.77)$$

where  $\mathbf{A}_a = \mathbf{A}_v$ . In this way, the desired  $[\dot{\boldsymbol{\omega}}_G^T \ \ddot{\tilde{\mathbf{p}}}_{sw_f}^T \ \ddot{\tilde{\mathbf{p}}}_{sw_h}^T]^T = \mathbf{A}_a^{-1} \mathbf{B}_a$ . It is noticed that this motion generator also enables the robot to turn with trotting gaits. In addition, we define the internal relationships among two swing legs. However, for practical applications, the relationships can be user-defined to accomplish more complex works. For example, when one obstacle is put in front of the robot, the front swing leg should lift off higher than the hind swing leg. The generated desired positions and velocities serve as the desired values for the compliance controllers in Section 5.4. The desired accelerations serve as the inputs of the prioritized multi-task torque controller in Section 5.4.2.

Until now, we have achieved the online each-leg motion respecting the contact constraints and the motion references, without leg-massless assumption and predefined leg trajectories. However, the centroidal angular motion is achieved by adding the extra internal legged motion relationships. In the future work, we will focus on the opposite logic, and define one suitable centroidal angular motion reference, then the legged internal motion relationships can be achieved on-line. In addition, the centroidal angular properties deal with the whole-robot stability [Lee 2012]. Therefore, in the future work, we will develop a general stability controller based on the robot angular momentum and one predictive swing-leg pick-up concept.

## 5.4 Whole-body Torque Controller

In this section, we define several impedance controllers to provide the robot with compliance abilities. We propose to integrate various impedance forces into one prioritized impedance torque which satisfies the task hierarchy and is 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 one optimization based feed-forward hierarchical torque controller which is developed by our previous work in [Du 2020f]. We control two tasks which are ranked in priority order from high to low as: the centroidal motion tracking task (signed as  $G$  with 6 DOF) and the swing-leg motion tracking task (signed as  $sw$  with 6 DOF for trotting and 3 DOF for walking). These two tasks should respect the constraints

relating to the contact situation and the parallel mechanism. The sum number of the whole-body DOF  $n = 22$  is all accounted for, shown in Tab. 5.3. It is noticed that the centroidal motion task can also be replaced by the floating-base motion task (signed as  $b$ ) which also accounts 6 DOF. Different with the conventional contact-consistent prioritized torque controller in [Park 2006], the constraints relating to the parallel mechanism in this chapter has the highest priority. Therefore, the whole-body torque controller in this section will handle this issue.

### 5.4.1 Prioritized Compliant Controller

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

Initially, one impedance controller is designed for the base motion as follows,

$$\Gamma_b = (\mathbf{W}_b^{-1} \mathbf{S}_u \mathbf{J}_b)^T [\mathbf{K}_b (\boldsymbol{\theta}_b^d - \boldsymbol{\theta}_b) + \mathbf{D}_b (\dot{\boldsymbol{\theta}}_b^d - \dot{\boldsymbol{\theta}}_b)], \quad (5.78)$$

where  $\mathbf{S}_u = [\mathbf{1}_{3 \times 3}, \mathbf{0}_{3 \times 3}]$ .  $\mathbf{J}_b = [\mathbf{1}_{6 \times 6} \quad \mathbf{0}_{6 \times (n-6)}]$  is the Jacobian for the base spatial motion.  $(\mathbf{K}_b, \mathbf{D}_b)$  denote the relative stiffness and damping matrices for the base angular motion.  $\mathbf{W}_b$  is defined in (??).

In addition, we define an impedance controller for the whole-body angular and translational momentums to track the robot centroidal motion as below,

$$\Gamma_G = \mathbf{J}_G^T \begin{bmatrix} \mathbb{I} + \mathbf{D}_{G_\omega} \bar{\mathbf{I}}_G (\mathbf{k}_G^d - \mathbf{k}_G) \\ \mathbf{K}_{G_v} (\mathbf{p}_G^d - \mathbf{p}_G) + \mathbf{D}_{G_v} (\mathbf{v}_G^d - \mathbf{v}_G) \end{bmatrix}, \quad (5.79)$$

where  $\mathbf{K}_{G_v} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{D}_{G_v} \in \mathbb{R}^{3 \times 3}$  denote the stiffness and damping matrices for the robot centroidal translational motion.  $\mathbf{D}_{G_\omega} \in \mathbb{R}^{3 \times 3}$  is the damping matrix for the centroidal angular motion tracking. The angular position of the whole-body CoM has no sense, and the integrability of the centroidal momentum model depends on precise conditions, detailed [Saccon 2017], and they are not guaranteed to be always satisfied for the case under study. Therefore, we applied the orientation control method in [Nava 2018] as follows,

$$\mathbb{I} = \mathbf{K}_{G_\omega} \bar{\mathbf{I}}_G \left[ \frac{1}{2} (\boldsymbol{\Omega}^T - \boldsymbol{\Omega}) \right]^V, \quad (5.80)$$

where  $\mathbf{K}_{G_\omega}$  is the stiffness matrix for the centroidal angular position tracking.  $\boldsymbol{\Omega} = {}^b \mathbf{R}_0^T ({}^b \mathbf{R}_0)^{ref}$ , and  $({}^b \mathbf{R}_0)^{ref}$  denotes the base orientation reference matrix. We define  $[\mathbf{A}]^V$  is the opposite function of one skew matrix  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$  as follows,

$$[\mathbf{A}]^V = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}^V = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \quad (5.81)$$

Another impedance controller for the centroidal height control relative to the base in the vertical direction of the inertial frame should be defined as below,

$$\Gamma_{Gb} = \mathbf{J}_{Gb}^T [\mathbf{K}_{Gb} (\mathbf{p}_{Gb}^d - \mathbf{p}_{Gb}) + \mathbf{D}_{Gb} (\dot{\mathbf{p}}_{Gb}^d - \dot{\mathbf{p}}_{Gb})], \quad (5.82)$$

where  $\mathbf{J}_{Gb} = \mathbf{S}_d \mathbf{J}_G - {}^b \mathbf{R}_0^T \mathbf{S}_d \mathbf{J}_b$  is the Jacobian for the centroidal translational motion relative to the base frame in the base vertical direction.  $\mathbf{S}_d = [\mathbf{0}_{3 \times 3}, \mathbf{1}_{3 \times 3}]$  is a selection matrix.  $\mathbf{K}_{Gb}$  and  $\mathbf{D}_{Gb}$  denote the relative stiffness and damping scalars.

The active compliance for legged suspension plays a significant role, therefore, for each swing-leg CoM motion, one impedance controller is established as below,

$$\Gamma_{sw_i} = \mathbf{J}_{p_i}^T [\mathbf{K}_{sw_i} (\tilde{\mathbf{p}}_{sw_i}^d - \tilde{\mathbf{p}}_{sw_i}) + \mathbf{D}_{sw_i} (\dot{\tilde{\mathbf{p}}}_{sw_i}^d - \dot{\tilde{\mathbf{p}}}_{sw_i})], \quad (5.83)$$

where  $\mathbf{J}_{p_i} = [\mathbf{0}_{3 \times 6} \quad \mathbf{S}_d \mathbf{J}_i^p]$ . In this impedance controller,  $i \in [sw_f, sw_h]$ .  $\mathbf{K}_{sw_i} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{D}_{sw_i} \in \mathbb{R}^{3 \times 3}$  denote respectively the stiffness and damping for the swing leg  $i$  CoM motion.

The above impedance controllers are used to track the external and internal motion references. However, when the robot arrives at the final target position of the external motion references, the robot may have the states different from the beginning state. Therefore, we enable the robot to trot in place and define another impedance controller for each swing-leg CoM motion as follows,

$$\mathbf{\Gamma}_{p_i} = (\mathbf{S}_{xy} \mathbf{J}_{p_i})^T \left[ \mathbf{K}_{p_i} \begin{pmatrix} \tilde{\mathbf{P}}_{i[x]}^d - \tilde{\mathbf{p}}_{i[x]} \\ \tilde{\mathbf{P}}_{i[y]}^d - \tilde{\mathbf{p}}_{i[y]} \end{pmatrix} + \mathbf{D}_{p_i} \begin{pmatrix} \dot{\tilde{\mathbf{P}}}_{i[x]}^d - \dot{\tilde{\mathbf{p}}}_{i[x]} \\ \dot{\tilde{\mathbf{P}}}_{i[y]}^d - \dot{\tilde{\mathbf{p}}}_{i[y]} \end{pmatrix} \right], \quad (5.84)$$

where  $\mathbf{K}_{p_i} \in \mathbb{R}^{2 \times 2}$  and  $\mathbf{D}_{p_i} \in \mathbb{R}^{2 \times 2}$  denote respectively the stiffness and damping for the swing leg  $i$  CoM motion,  $i \in [sw_f, sw_h]$ .  $\tilde{\mathbf{P}}_{i[x]}^d$  and  $\tilde{\mathbf{P}}_{i[y]}^d$  denote the desired leg  $i$  CoM position in the base frame corresponding the robot beginning state and they have constant values,  $\dot{\tilde{\mathbf{P}}}_{i[x]}^d = \dot{\tilde{\mathbf{P}}}_{i[y]}^d = 0$ . This impedance controller can imitate the tendon on animals and enable the robot to trot in place and recover to the beginning state. In addition, we set the stiffness and damping matrices with small values to reduce their influences on the other impedance controllers.

According to the internal legged motion relationships in Section 5.3.9, we define two impedance controllers on normal terrains (e.g. flat terrain) as follows,

$$\mathbf{\Gamma}_f = \mathbf{J}_f^T [\mathbf{K}_f \mathbf{e}_f + \mathbf{D}_f \dot{\mathbf{e}}_f], \quad (5.85a)$$

$$\mathbf{\Gamma}_h = \mathbf{J}_h^T [\mathbf{K}_h \mathbf{e}_h + \mathbf{D}_h \dot{\mathbf{e}}_h], \quad (5.85b)$$

where  $(\mathbf{K}_f, \mathbf{D}_f)$  and  $(\mathbf{K}_h, \mathbf{D}_h)$  denote the relative stiffness and damping matrices.  $(\mathbf{J}_f, \mathbf{J}_h)$  and  $(\mathbf{e}_f, \mathbf{e}_h)$  are defined as below,

$$\mathbf{J}_f = \begin{bmatrix} \mathbf{S}_x (\mathbf{J}_{p_{sw_f}} + \mathbf{J}_{p_{st_h}}) \\ \mathbf{S}_y (\mathbf{J}_{p_{sw_f}} + \mathbf{J}_{p_{st_f}}) \\ \mathbf{S}_z (\mathbf{J}_{p_{sw_f}} - \mathbf{J}_{p_{sw_h}}) \end{bmatrix}, \quad (5.86a)$$

$$\mathbf{J}_h = \begin{bmatrix} \mathbf{S}_x (\mathbf{J}_{p_{sw_h}} + \mathbf{J}_{p_{st_f}}) \\ \mathbf{S}_y (\mathbf{J}_{p_{sw_h}} + \mathbf{J}_{p_{st_h}}) \\ \mathbf{S}_z (\mathbf{J}_{p_{sw_h}} - \mathbf{J}_{p_{sw_f}}) \end{bmatrix}, \quad (5.86b)$$

$$\mathbf{e}_f = \begin{bmatrix} -\mathbf{S}_x (\tilde{\mathbf{p}}_{sw_f} - \tilde{\mathbf{P}}_{sw_f}^d) - \mathbf{S}_x (\tilde{\mathbf{p}}_{st_h} - \tilde{\mathbf{P}}_{st_h}^d) \\ -\mathbf{S}_y (\tilde{\mathbf{p}}_{sw_f} - \tilde{\mathbf{P}}_{sw_f}^d) - \mathbf{S}_y (\tilde{\mathbf{p}}_{st_f} - \tilde{\mathbf{P}}_{st_f}^d) \\ -\mathbf{S}_z (\tilde{\mathbf{p}}_{sw_f} + \tilde{\mathbf{p}}_{sw_h}^d) \end{bmatrix}, \quad (5.86c)$$

$$\mathbf{e}_h = \begin{bmatrix} -\mathbf{S}_x (\tilde{\mathbf{p}}_{sw_h} - \tilde{\mathbf{P}}_{sw_h}^d) - \mathbf{S}_x (\tilde{\mathbf{p}}_{st_f} - \tilde{\mathbf{P}}_{st_f}^d) \\ -\mathbf{S}_y (\tilde{\mathbf{p}}_{sw_h} - \tilde{\mathbf{P}}_{sw_h}^d) - \mathbf{S}_y (\tilde{\mathbf{p}}_{st_h} - \tilde{\mathbf{P}}_{st_h}^d) \\ -\mathbf{S}_z (\tilde{\mathbf{p}}_{sw_h} + \tilde{\mathbf{p}}_{sw_f}^d) \end{bmatrix}, \quad (5.86d)$$

where  $sw_f$ ,  $sw_h$ ,  $st_f$  and  $st_h$  represent the swing-front, swing-hind, stance-front and stance-hind leg index, and  $\mathbf{J}_{p_i}$  is defined in (5.83),  $\tilde{\mathbf{P}}_i^d$  is defined in (5.84).

The impedance controllers should be integrated into the operational prioritized control framework and they should be consistent with task priorities. Therefore, we define  $\mathbf{\Gamma} \in \mathbb{R}^n$  to include all impedance controllers as below,

$$\mathbf{\Gamma} = (\mathbf{J}_{G|c|m}^T \bar{\mathbf{J}}_{G|c|m}^T + \mathbf{J}_{sw|G|c|m}^T \bar{\mathbf{J}}_{sw|G|c|m}^T) \sum \mathbf{\Gamma}_k, \quad (5.87)$$

where  $\sum \mathbf{\Gamma}_k$  denotes the sum of all the impedance controllers defined in this subsection.  $\sum \mathbf{\Gamma}_k$  can also integrate any other user-defined impedance controllers.  $\mathbf{J}_{G|c|m}$  and  $\mathbf{J}_{sw|G|c|m}$  represent the Jacobians for the centroidal motion and the swing-leg motion tasks respectively as follows,

$$\mathbf{J}_{G|c|m} = \mathbf{J}_G \mathbf{N}_{c|m} = \mathbf{J}_G (\mathbf{N}_m - \bar{\mathbf{J}}_{c|m} \mathbf{J}_{c|m}), \quad (5.88a)$$

$$\mathbf{J}_{sw|G|c|m} = \mathbf{J}_{sw} \mathbf{N}_{G|c|m} = \mathbf{J}_{sw} (\mathbf{N}_{c|m} - \bar{\mathbf{J}}_{G|c|m} \mathbf{J}_{G|c|m}), \quad (5.88b)$$

Table 5.3: Task Control Hierarchy

Task	Name	Trot	Walk	Stance
1	Mechanism constraints $m$	4 DOF	4	4
	Contact constraints $c$	6 DOF	9	12
	Centroidal motion $G$ or Floating base motion $b$	6 DOF	6	6
	Swing-leg motion $sw$	6 DOF	3	0

where  $\mathbf{J}_{sw}$  is the swing-leg motion Jacobian which includes two slack  $\mathbf{J}_{p_i}, i \in [sw_f, sw_h]$  for the trotting mode, and  $\mathbf{J}_{sw}$  includes only one  $\mathbf{J}_{p_i}, i = sw$  for the walking mode.  $\mathbf{N}_{c|m}$  denotes the null-space projector for the contact and mechanism constraints, and  $\mathbf{N}_{G|c|m}$  is the null-space projector combining the centroidal motion tracking task and the two constraints. The subscript  $c|m$  means that the contact constraint task cannot conflict the mechanism constraint. The subscript  $G|c|m$  means that the CoM tracking task cannot conflict the two constraints. Similarly, the subscript  $sw|G|c|m$  means that the swing-leg motion task should be in the null-space of the CoM tracking task and the two constraints. Each  $\bar{\mathbf{J}} = \mathbf{M}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{M}^{-1} \mathbf{J}^T)^{-1}$  denotes the dynamically-consistent inverse of each task Jacobian  $\mathbf{J}$ . The null-space based inverse dynamics can be referred to [Sentis 2007].  $\mathbf{\Gamma}$  is the prioritized impedance controller, detailed in Chapter 3.  $\mathbf{\Gamma}$  will not conflict the two kinematics constraints and respect the task hierarchy. In addition, the subscript  $G$  in (5.87) and (5.88) can be replaced by  $b$  if the floating base motion task is chosen as the first task.

### 5.4.2 Optimization based Torque Controller

For the whole-body torque controller, we use the torque controller P-HQP in Chapter 3. This technique presents many advantages for our legged locomotion. All tasks are projected in the null-space of the contact constraints. The prioritized impedance controller was proposed to be integrated into this hybrid framework, which influences the designed tasks in a decoupled way. For example, the impedance controller for the similar swing-leg motion cannot conflict the designed task hierarchy. The task control hierarchies for the walking and trotting locomotion modes are defined in Table 5.3. In this subsection, the index  $i$  means the hierarchy index.

It is noticed that for our robot, the constraints relating to the parallel mechanism has the highest priority. This is different from the state of the art that considers the contact constraint with highest priority. To be more general, we derive the dynamic properties relating to the hierarchy  $i$  operational-space task as follows,

$$\mathbf{\Lambda}_{i|pre(i)} = (\mathbf{J}_{i|pre(i)} \mathbf{M}^{-1} \mathbf{J}_{i|pre(i)}^T)^{-1}, \quad (5.89a)$$

$$\bar{\mathbf{J}}_{i|pre(i)}^T = \mathbf{\Lambda}_{i|pre(i)} \mathbf{J}_{i|pre(i)} \mathbf{M}^{-1}, \quad (5.89b)$$

$$\boldsymbol{\mu}_{i|pre(i)} = \mathbf{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{C} + \mathbf{\Lambda}_{i|pre(i)} \mathbf{J}_i \bar{\mathbf{J}}_{c|m} \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \mathbf{\Lambda}_{i|pre(i)} \dot{\mathbf{J}}_i \dot{\mathbf{q}}, \quad (5.89c)$$

$$\boldsymbol{\rho}_{i|pre(i)} = \mathbf{\Lambda}_{i|pre(i)} \mathbf{J}_i \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{G}, \quad (5.89d)$$

where  $\mathbf{J}_{i|pre(i)} = \mathbf{J}_i \mathbf{N}_{pre(i)}$  is the prioritized operational space Jacobian for task  $i$ , which are in the null-space of previous tasks, represented as  $pre(i)$ .  $\mathbf{N}_{pre(i)}$  is the null-space by combining all tasks above level  $i$  which can be calculated recursively as  $\mathbf{N}_{pre(i+1)} = \mathbf{N}_{pre(i)} - \bar{\mathbf{J}}_{i|pre(i)} \mathbf{J}_{i|pre(i)}$ . Furthermore,  $\mathbf{N}_{pre(1)} = \mathbf{N}_{c|m}$ . For our application,  $K = 2$ ,  $\mathbf{J}_1 = \mathbf{J}_G$  and  $\mathbf{J}_2 = \mathbf{J}_{sw}$ . It is noticed that  $\mathbf{J}_1 = \mathbf{J}_b$  if the tracking of the floating-base motion is selected as the first task. In addition,  $\mathbf{\Lambda}_{i|pre(i)}$  represents the task  $i$  operational space inertia.  $\boldsymbol{\mu}_{i|pre(i)}$  and  $\boldsymbol{\rho}_{i|pre(i)}$  denote the the Coriolis & centrifugal force, and the gravity force in the task  $i$  operational space respectively. The dynamics properties for the designed

two tasks are listed as follows.

$$\Lambda_{G|c|m} = (\mathbf{J}_{G|c|m} \mathbf{M}^{-1} \mathbf{J}_{G|c|m}^T)^{-1}, \quad (5.90a)$$

$$\bar{\mathbf{J}}_{G|c|m}^T = \Lambda_{G|c|m} \mathbf{J}_{G|c|m} \mathbf{M}^{-1}, \quad (5.90b)$$

$$\boldsymbol{\mu}_{G|c|m} = \Lambda_{G|c|m} \mathbf{J}_G \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{C} + \Lambda_{G|c|m} (\mathbf{J}_G \bar{\mathbf{J}}_{c|m} \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \dot{\mathbf{J}}_G \dot{\mathbf{q}}), \quad (5.90c)$$

$$\boldsymbol{\rho}_{G|c|m} = \Lambda_{G|c|m} \mathbf{J}_G \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{G}, \quad (5.90d)$$

where it is noticed that the subscript  $G$  can be replaced by  $b$  if tracking the floating-base motion is the first task. Similarly, the dynamic properties for the swing-leg motion tracking task are derived as below,

$$\Lambda_{sw|G|c|m} = (\mathbf{J}_{sw|G|c|m} \mathbf{M}^{-1} \mathbf{J}_{sw|G|c|m}^T)^{-1}, \quad (5.91a)$$

$$\bar{\mathbf{J}}_{sw|G|c|m}^T = \Lambda_{sw|G|c|m} \mathbf{J}_{sw|G|c|m} \mathbf{M}^{-1}, \quad (5.91b)$$

$$\boldsymbol{\mu}_{sw|G|c|m} = \Lambda_{sw|G|c|m} \mathbf{J}_{sw} \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{C} + \Lambda_{sw|G|c|m} (\mathbf{J}_{sw} \bar{\mathbf{J}}_{c|m} \dot{\mathbf{J}}_c \dot{\mathbf{q}} - \dot{\mathbf{J}}_{sw} \dot{\mathbf{q}}), \quad (5.91c)$$

$$\boldsymbol{\rho}_{sw|G|c|m} = \Lambda_{sw|G|c|m} \mathbf{J}_{sw} \mathbf{M}^{-1} \mathbf{N}_{c|m}^T \mathbf{G}. \quad (5.91d)$$

Since the constraints relating to the parallel mechanism induce a change into the dynamics model, the operational-space properties in (5.89) should be applied to achieve each task-acceleration extraction  $\ddot{\mathbf{x}}_i$  which is used to build the cost function of P-HQP as follows,

$$Q_i(\boldsymbol{\tau}_i) = \|\ddot{\mathbf{x}}_i^d - \ddot{\mathbf{x}}_i\|^2 + \varepsilon \|\boldsymbol{\tau}_i\|^2, \quad (5.92)$$

where  $Q_i(\boldsymbol{\tau}_i)$  is the cost function and  $\boldsymbol{\tau}_i$  is the optimized torque for hierarchy  $i$ .  $\ddot{\mathbf{x}}_i^d$  is the desired task  $i$  acceleration,  $\ddot{\mathbf{x}}_1^d = \dot{\mathbf{v}}_G^d$  and  $\ddot{\mathbf{x}}_2^d = \dot{\mathbf{v}}_{sw}^d$ , and they are achieved in the motion generator. The control objective is regularized by a small value  $\varepsilon$ , which ensures positive definiteness of the objective Hessian. The common inequality constraints for hierarchy  $i$  optimization include the joint torque/position/velocity limits, the contact unilateral constraint and the friction cone as below,

$$\boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_{act_i} \leq \boldsymbol{\tau}_{max} \quad (5.93a)$$

$$\underline{\mathbf{q}}_{min} \leq \underline{\mathbf{q}}_{|k+1} = \underline{\mathbf{q}}_{|k} + \underline{\dot{\mathbf{q}}}_{|k} \delta t + \frac{1}{2} \underline{\ddot{\mathbf{q}}}_{|k} \delta t^2 \leq \underline{\mathbf{q}}_{max} \quad (5.93b)$$

$$\underline{\dot{\mathbf{q}}}_{min} \leq \underline{\dot{\mathbf{q}}}_{|k+1} = \underline{\dot{\mathbf{q}}}_{|k} + \underline{\ddot{\mathbf{q}}}_{|k} \delta t \leq \underline{\dot{\mathbf{q}}}_{max} \quad (5.93c)$$

$$\mathbf{u}_c \cdot \mathbf{F}_c \geq 0, \quad (5.93d)$$

$$|\mathbf{t}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c) \quad (5.93e)$$

$$|\mathbf{b}_c \cdot \mathbf{F}_c| \leq \frac{\mu}{\sqrt{2}} (\mathbf{u}_c \cdot \mathbf{F}_c), \quad (5.93f)$$

where  $\boldsymbol{\tau}_{act_i}$  is the actuated torque combining the optimization variable in hierarchy  $i$ ,  $\boldsymbol{\tau}_i$ , and the variables which have be optimized in hierarchies  $< i$ ,  $\boldsymbol{\tau}_t$ , ( $t \in [1, \dots, i-1]$ ) as follows,

$$\boldsymbol{\tau}_{act_i} = \left[ \mathbf{W} \mathbf{N}_{pre(i)}^T \mathbf{S}^T \right] \boldsymbol{\tau}_i + \left[ \mathbf{W} \sum_{t=1}^{i-1} \mathbf{N}_{pre(t)}^T \mathbf{S}^T \boldsymbol{\tau}_t \right], \quad (5.94a)$$

$$\mathbf{W} = (\mathbf{S} \mathbf{N}_{c|m} \mathbf{M}^{-1} \mathbf{S}^T)^+ \mathbf{S} \mathbf{N}_{c|m} \mathbf{M}^{-1}, \quad (5.94b)$$

where the superscript  $+$  means the Moore-inverse method.  $\mathbf{W}$  is a mapping matrix and enables the calculated torque to be consistent with the kinematics constraints, and its original version is defined in [Sentis 2010] without kinematics constraints relating to parallel mechanisms. The quadratic forms of the contact force and the actuated-joint acceleration are extracted according to the optimized variable  $\boldsymbol{\tau}_i$  as follows,

$$\mathbf{F}_c = \mathbf{A}_{c_i}^T \boldsymbol{\tau}_i + \mathbf{B}_{c_i}, \quad (5.95a)$$

$$\underline{\ddot{\mathbf{q}}} = \mathbf{A}_{q_i}^T \boldsymbol{\tau}_i + \mathbf{B}_{q_i}, \quad (5.95b)$$

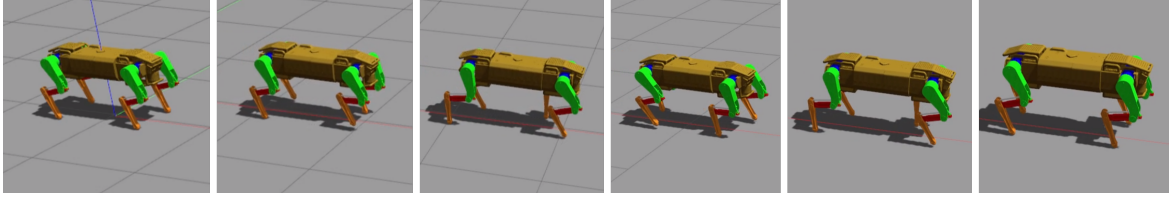


Figure 5.11: Snapshots of the forward trotting for  $15.2m$  with  $0.8m/s$  velocity. The robot squats down from the base height  $0.53m$  to  $0.5m$ , then accelerates from zero speed to  $0.8m/s$  within  $1s$ . Then the robot trots with constant speed  $0.8m/s$  for  $18s$ , then decelerates to zero speed using the final  $1s$ .

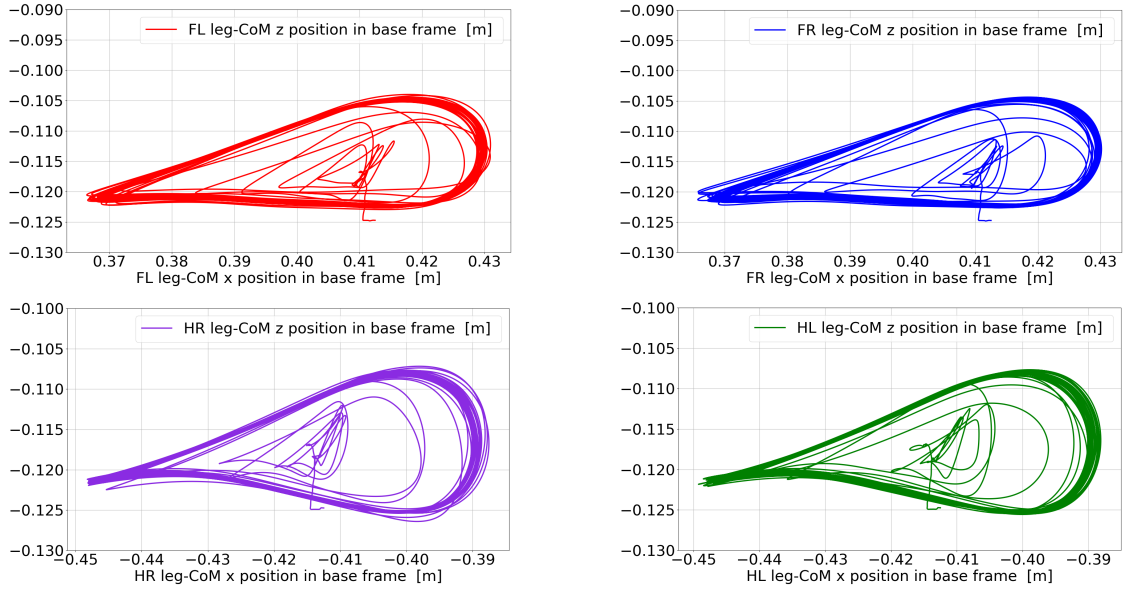


Figure 5.12: Four leg-CoM motions in the sagittal plane when the robot trotting forward for 80 steps. This figure includes the acceleration and constant-speed periods. The curves inside the stable loops represent the transition period when the robot accelerates.

where  $(A_{c_i}, B_{c_i})$  and  $(A_{q_i}, B_{q_i})$  are derived as below,

$$A_{c_i}^T = -\bar{J}_{c|m}^T S^T W N_{pre(i)}^T S^T, \quad (5.96a)$$

$$B_{c_i} = \mu_{c|m} + \rho_{c|m} + \bar{J}_{c|m}^T (\Gamma - S^T W \sum_{t=1}^{i-1} N_{pre(t)}^T S^T \tau_t), \quad (5.96b)$$

$$A_{q_i}^T = \underline{M}^{-1} \bar{J}_{a|m}^T (\bar{J}_c^T A_{c_i}^T + S^T W N_{pre(i)}^T S^T), \quad (5.96c)$$

$$B_{q_i} = \underline{M}^{-1} \bar{J}_{a|m}^T (\bar{J}_c^T B_{c_i} + S^T W \sum_{t=1}^{i-1} N_{pre(t)}^T S^T \tau_t) - \underline{M}^{-1} (\underline{C} + \underline{G} + \bar{J}_{a|m}^T \Gamma), \quad (5.96d)$$

We substitute (5.94) and (5.95) into (5.93) to constitute the compatible constraints for hierarchy  $i$ . After the optimization of all hierarchies, the final actuated torque  $\tau_{act\kappa}$  can be achieved, shown in Fig. 5.5. The contact force can be estimated by substituting  $\tau_{act}$  into (5.20).

### 5.4.3 Lateral Push-recovery Reaction

As we know, the robot cannot resist to the lateral large-force push because of actuated-power limits. The optimization based torque controller in Section 5.4.2 cannot generate the required actuated torque





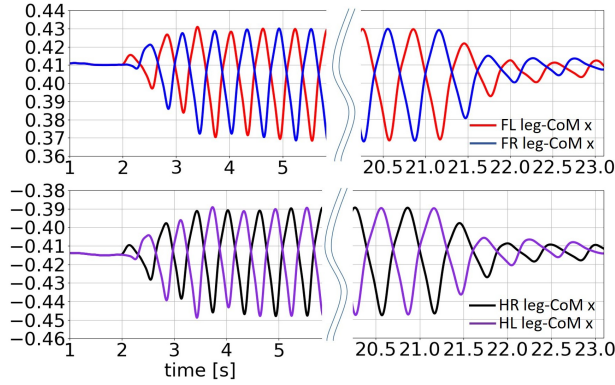


Figure 5.14: Transition of the legged motion in  $x_b$  axis when the robot accelerates and decelerates in the forward direction. After  $t = 23s$ , the robot trots in place in the final target position.

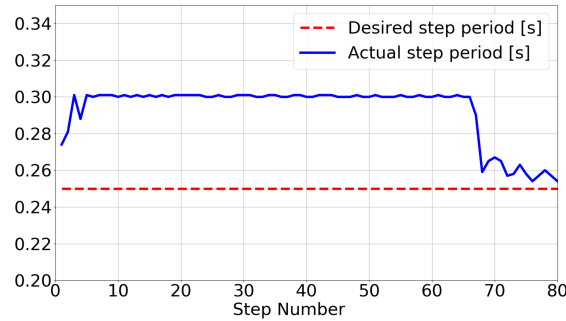


Figure 5.15: Forward-trotting step period record for 80 steps.

### 5.5.1 Trotting Forward

In this scenario, we simulate our virtual robot to trot forward on the flat terrain for  $15.2m$  using 80 steps. In addition,  $\sigma$  in the state machine is set to be 1.2 which means that the swing-leg indexes are enforced to be changed when the actual step period reaches 120% of the desired value. From the snapshots in Fig. 5.11 and the leg-CoM position in the beginning and ending of the whole process in Fig. 5.14, it is noticed that the robot is able to trot smoothly from speed zero to  $0.8m/s$  without defining the step length and the transition among different speeds. Compared to the state of the art, the smooth transition between different speeds is a major advantage of our new locomotion principle and the corresponding legged motion generator, detailed in Section 5.3. At low speeds, the leg-CoM position along the  $x_b$  axis in the base frame has a small-amplitude change. At the constant speed  $0.8m/s$ , each leg CoM experiences a stable change, c.f. the middle phrases in Fig. 5.14. The front and rear leg-CoM trajectories experience different stance motions w.r.t. the base, due to the fact that the actual base orientation and the whole-body motion are influenced by the mutual influences between the impact forces and the optimized torque, and latter one is generated by the hierarchical optimization based torque controller by combining the feed-forward acceleration-reference tracking and the feedback general impedance controllers, especially in (5.84) and (5.85). In Fig. 5.12 and Fig. 5.13, we can see that each leg CoM or foot reaches a stable cycle after the legs pass the transition period. The foot-position loops in Fig. 5.13 follow the relationships with the leg CoM-position loops in Fig. 5.12, detailed in the leg design part in Section 5.2.1.2.

The desired step period  $T_{sw_i} = 0.25s$  and the actual step period is always longer than the desired value, c.f. Fig. 5.15. There are three reasons: The contact point is estimated at the middle tip of foot which is not accurate for the ball-like foot. The base has rotational angles when trotting with two stance legs. The impact force in each step period results in the fluctuation of the base motion in the vertical direction. The step period tracking can be improved by increasing the impedance controllers, however, it will also lead to larger impact force and more fluctuation of the base motion. In the future work, the

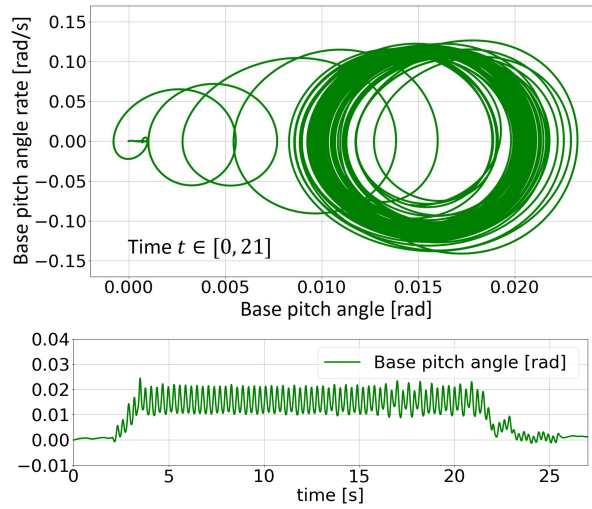


Figure 5.16: [Up]: The periodic cycles by one filter. [Down]: Actual base pitch angle.

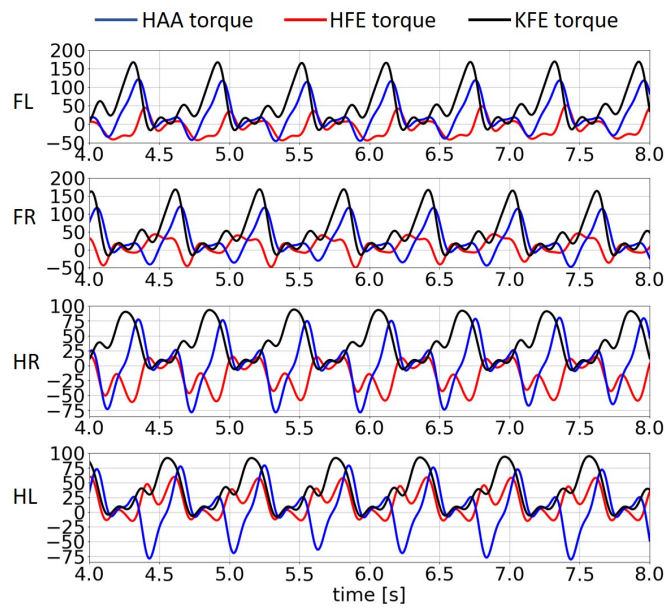


Figure 5.17: The generated torque for four-leg joints [N/m].

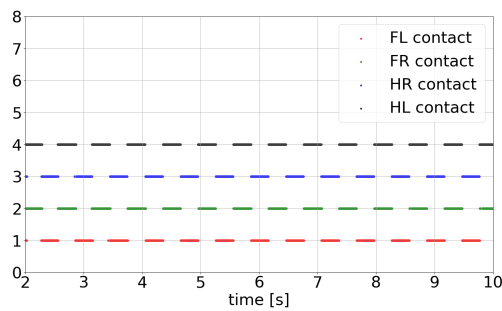


Figure 5.18: The detected contact states of four legs when trotting forward during  $t \in [2, 10]$ .

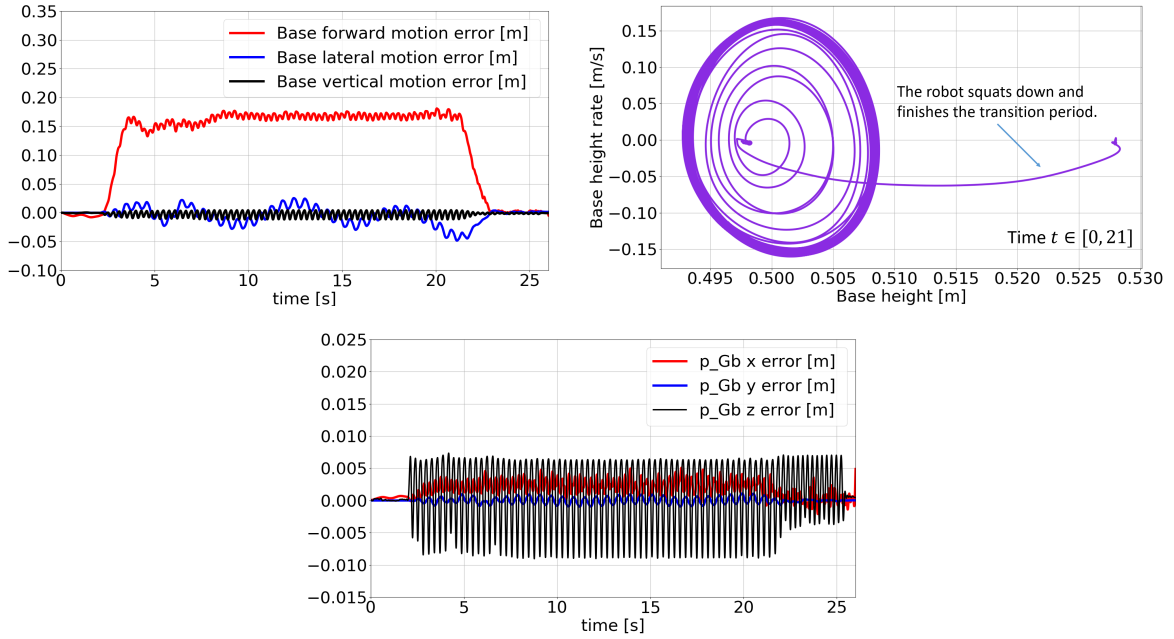


Figure 5.19: Base and robot-CoM tracking performance for forward trotting motion.

vision function will be applied and one more smooth  $\mathbf{p}_{Gb}$  trajectory will be generated on-line which will be tracked by impedance controllers with bigger stiffness and damping. In addition, Fig. 5.18 shows the contact states of four legs during  $t \in [2, 10]$ , at some specific timings in this trotting mode. There are three contact points, which means that two legs land at different timings and the walking mode is applied instead of the trotting mode until four legs all touch the ground, detailed in the state machine in Section 5.3.5.

In the running process, the dynamics model is discontinuous due to the frequent changes in the dimension of the contact Jacobian in (5.4). The sudden contact forces are partially compensated by the impedance controllers, and partially transferred to the base resulting in rotational motion of the base. We show the pitch angle along the whole process: it is very small and satisfactory by using several various impedance controllers which are projected to the hierarchical torque controller by respecting the contact constraints and the task hierarchy. The generated torque for each-leg actuated joints are shown in Fig. 5.17. In addition, the smooth transition property can also be seen in the pitch-angle tracking performance. The stable-loop property can be observed in the generated torque in Fig. 5.17 in which the curves are selected during the constant-speed trotting process and each-joint torque experiences periodically.

Furthermore, we show the tracking performances of the base translational motion and the relative motion between the base and the robot CoM in Fig. 5.19. Along the whole process, the tracking performance is quite satisfactory. However, the base forward motion experiences a little delay, because for our new locomotion principle, the base or the centroidal spatial motion serves as the reference, then the stance legs follow the base motion. Although the desired swing-leg velocity and acceleration are generated using the desired stance-leg velocity and acceleration respectively, the swing-leg position is achieved using the actual stance-leg position whose contact points are not accurately estimated, resulting in the stance legs having a little delay. Since we use the relative motion as the reference in the legged motion generator and  $\mathbf{p}_{Gb_x} = 0$ , then the desired swing-leg positions suffer a little delay and the impedance controllers relating to the legged motion pull the base behind a little. However, the base delay influence can be improved by adjusting the stiffness and damping matrices in the impedance controllers relating to the base motion. In addition, the base height also fluctuates during step period due to the sudden contact forces. We can see in Fig 5.19 that the base height reaches periodic loops after the robot squats down and finishes the transition period.

In addition, it is noticed that in the simulation video, the robot recovers to its beginning state without defining the step number to be odd or even, because the impedance controller in (5.84) pulls the legs to the beginning state. However, it also causes a little delay of the base motion and influences the symmetric motion in the running process, which can be viewed in the Y-axis values in Fig. 5.14 and the X-axis values in Fig. 5.12 and Fig. 5.13. Compared to its drawbacks, the advantage is more apparent without considering how the robot stops.

### 5.5.2 Trotting Lateral, Turning in Place and Push-recovery

The new locomotion principle and legged motion generator also enable the robot to trot in the lateral direction, turning and recovering from external push force. Some of its characteristics are similar with the trotting mode in the forward direction and are not plotted in this subsection. Initially, the robot is allowed to trot in the lateral direction for  $2.5m$  with  $0.25m/s$  by given the base lateral motion reference and the snapshots are shown in Fig. 5.20. The legged motion can also be generated on-line. The tracking results in the lateral direction in Fig. 5.21 also show similar performances as the forward trotting in Fig. 5.19.

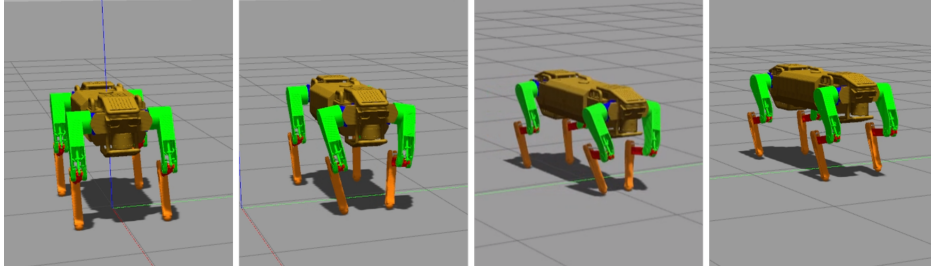


Figure 5.20: Snapshots of the lateral trotting with  $0.25m/s$ .

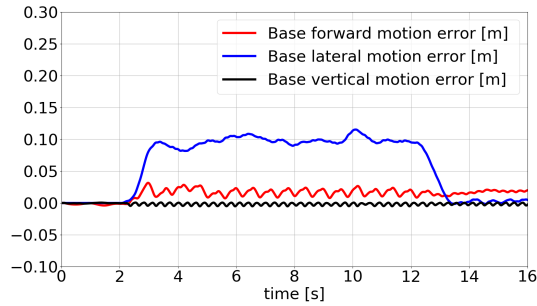


Figure 5.21: Base tracking performance for the lateral trotting.

Apart from the translational-motion capabilities, our approach also enables our robot to behave turn-trotting motion because the internal legged motion relationships defined in Section 5.3.9 can also handle the base turning motion. In this scenario, the desired step period is  $T_{sw_i} = 0.28s$ , the robot first turns in the counterclockwise direction for  $\frac{3\pi}{4}rad$ , then in the opposite direction for  $\pi rad$ , shown in Fig. 5.22. The stance legs follow the base rotational motion, the swing-leg motion are automatically generated. The tracking results of the base roll, pitch and yaw angles are depicted in Fig. 5.23 which are very small along the whole process by using our prioritized impedance controller based hierarchical whole-body torque control.

In addition, the proposed legged motion generator can also handle the push-recovery motion, shown in Fig. 5.24[Left] in which the red-thin line represent the  $x_0$  axis of the inertial frame. The robot is simulated to trot forward with  $0.5m/s$  and during  $t \in [9, 12]$ , an external virtual lateral force in Gazebo is applied onto the base, then the robot begins to trot in the lateral direction while trotting forward. We

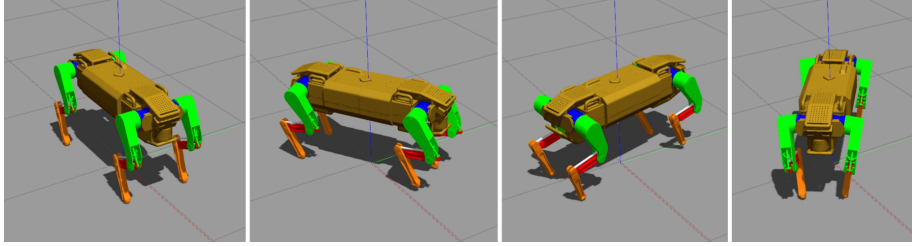


Figure 5.22: Snapshots of the turn trotting in place with  $\frac{\pi}{8} \text{rad/s}$  using 64 steps. The robot trot to turn in the counterclockwise and clockwise directions.

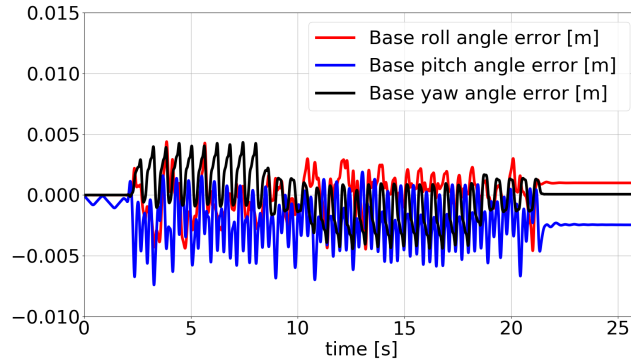


Figure 5.23: Base angular-motion tracking performance for turn-trotting motion.

can see in Fig. 5.24[Left] that the robot cross over the red  $x_0$  axis. Then the robot trots back to track the base translational motion reference by itself. We show the tracking performance of the base lateral position and the roll angle in Fig. 5.24[Right] in which the inside figure denotes the push-force profile. The simulation video shows the apparent legged reaction during the push period.

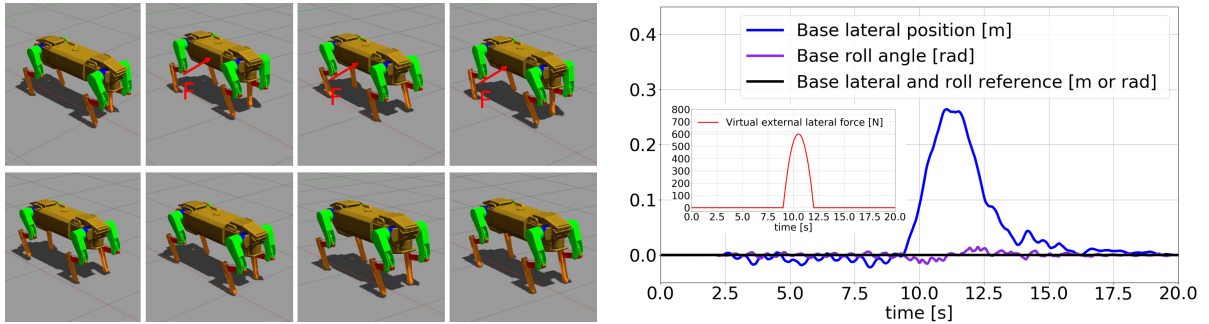


Figure 5.24: [Left]: Snapshots of the push recovery by trotting in place with  $0.5 \text{m/s}$  as far as  $6.5 \text{m}$ . [Right]: Push recovery when the robot trots forward with  $0.5 \text{m/s}$ . The small figure inside denotes the virtual external force profile.

Until now, we show the abundant capabilities of our new locomotion principle and the proposed legged motion generator to achieve the trotting mode to track the base translational motion, turning motion, and the ability to resist disturbances.

### 5.5.3 Trotting on Rough Terrain

In the final scenario, the robot is simulated to trot on relative rough terrain with  $0.5 \text{m/s}$  for 210 steps, shown in Fig. 5.25. The rough terrain contains three parts: the first road in the  $x_0$  axis has five small stairs, the second road with the angle of  $45^\circ$  to the  $x_0$  axis are configured with two symmetric bumps, the



third road is along the  $y_0$  axis with unsymmetrical 3-left and 2-right bumps. The tracking performance

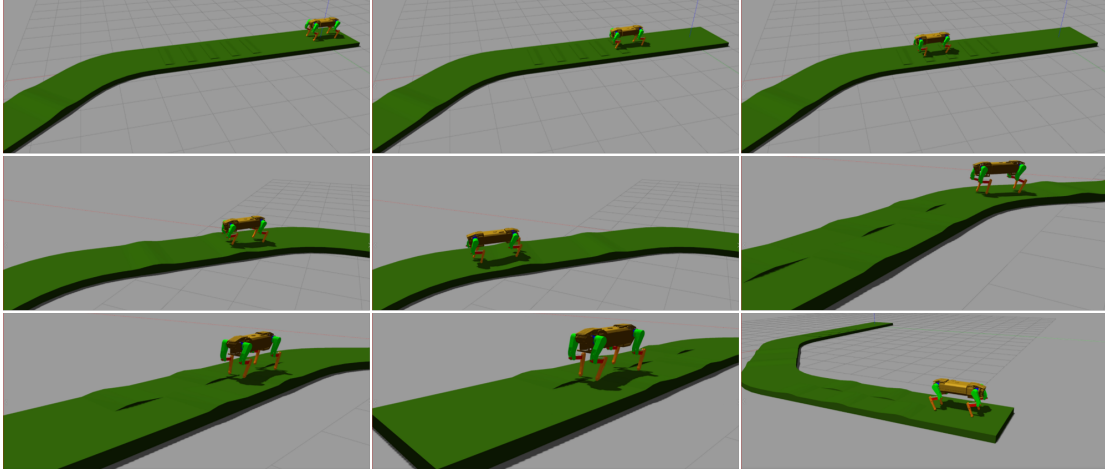


Figure 5.25: Snapshots of trotting on rough terrain with  $0.5m/s$  using 210 steps.

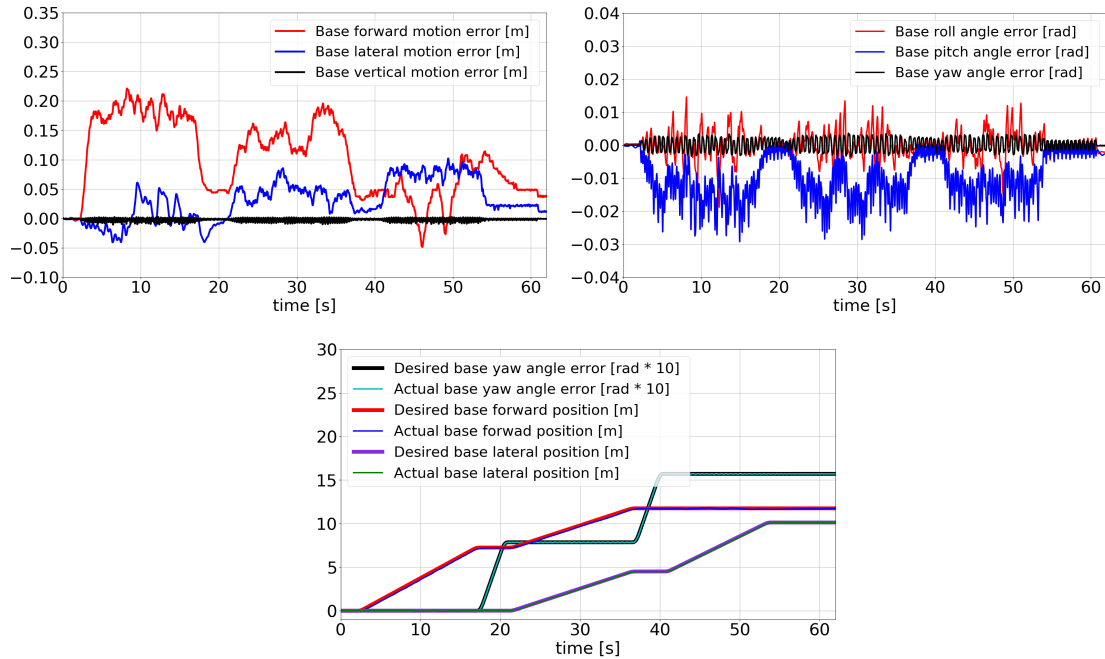


Figure 5.26: Base spatial motion tracking performance for the trotting mode on rough terrain. [Up]: base translational motion tracking error. [Middle]: base angular motion tracking error. [Down]: selected desired and actual motion comparison along the whole process.

of the base translational and angular motions are depicted in Fig. 5.26. It is noticed that the emerging big errors happen when the robot crosses over the rough terrains,  $t \in [3s, 7s]$  for the stairs,  $t \in [23s, 36s]$  for the symmetric bumps, and  $t \in [41s, 54s]$  for the unsymmetrical bumps, shown in Fig. 5.26[Up] and Fig. 5.26[Middle]. Apart from the compliance from the torque controller, the adaptive motion reference  $\mathbf{p}_{Gb_z}$  in Section 5.3.4 also enables the swing legs to continue to go down until the detection of the contact signal. The Fig. 5.26[Down] depicts the desired and actual base motion comparison along  $(x_0, y_0)$  axes and around  $z_0$  axis. It shows that desired and actual base trajectories along the whole process are almost coincident. Although the robot has fluctuation in the lateral direction and has delay in the forward direction, the robot succeeds to cross over the three unknown roads. In the simulation video, by given



the base motion references, we can see that the robot adjust its configuration and legged trajectory on-line by itself by using our legged motion generator and compliant torque controller.

## 5.6 Summary

This chapter proposes a new legged locomotion principle and one legged motion generator for a quadruped robot. The robot leg is new designed with parallelogram mechanisms to enable the robot with more symmetric mass distribution and dynamics properties and the generalized dynamics model is built. Then the centroidal momentum and dynamics models are established using each leg centroidal motion instead of each leg-body motion, and each leg centroidal properties are derived. The motion generator is proposed which is much closer to real quadruped-animal locomotion principle in terms of the automatic gait generation. The stance legs follow the base motion using contact-constraint consistent torque controller. The swing legs compensate the influences from the stance legs to the centroidal motion. The desired legged position, velocity and accelerations are achieved using the mass distribution model, the modified centroidal momentum and dynamics models, and the kinematics contact constraints. By combining the whole-body kinematics model and the kinematics constraints relating to the contact constraints and the parallelogram mechanism, our new locomotion principle and generator enable the robot to trot and walk without pre-defined legged-motion trajectories. When two swing legs land in different timings, the walking gait is applied. The proposed state machine is developed for switching the locomotion gaits between walking and trotting, which also serves the torque controller and determines the dimension of the legged control states and the contact Jacobian. Therefore, our new locomotion principle integrates the system design, motion generation and torque control together. This one common framework synthesizes several locomotion gaits using only one control framework. The validation is performed through advanced numerical simulations using our quadruped robot in the multi-body multi-contact dynamic engine. The robot succeeds to trot in the forward and lateral directions, turn, handle push-recovery motion, as well as trot on the relative rough terrain.

# Conclusion and Future Works

---

In this thesis, we initially propose and improve four kinds of whole-body torque controllers. Then we develop two motion generators for the quadruped-on-wheel robot, *TowrISIR*. Finally, we propose to generate the trotting-mode locomotion for the quadruped robot, *QuadISIR* without the leg-massless assumption or pre-defined leg trajectory. The proposed approaches are verified through abundant simulation scenarios. In this chapter, we summarize these works and present the future perspectives.

## 6.1 Conclusions

Focusing first on the torque controller, we improve four kinds of whole-body torque controllers by proposing new dynamics models and one concept, “Prioritized Impedance Controller” (PIC). PIC enables various impedance control forces to be consistent and compatible with multi-task hierarchies. The concept can handle more general and various impedance controllers, including the ones built directly on specific designed tasks, and ones indirectly acting on the whole-body behaviors. Regarding the pseudo-inverse based dynamics method and null-space based weighted optimization solver, we propose to modify the generalized dynamics model to include the PIC. In contrast with the conventional weighted QP, we first propose to introduce priorities in the weighted scheme, in the same way, hierarchies exist among several subtasks. The proposed PIC decouples the influences from various impedance forces to the designed tasks in the mathematical way. The modified dynamics model enables to integrate this new weighted scheme into the hierarchical control framework. For the optimization based hierarchical techniques (including strict hierarchical convex optimization and a hybrid hierarchical-weighted scheme), another new dynamics model is proposed which also contains the PIC and is more general for hierarchical optimization. The modified or new dynamics model embodies the multi-level control hierarchy and enables the achievement of the more general and complete hierarchical optimization solvers, both for the task acceleration extraction and the same number of constraints. The detailed algorithm is developed which is more general and can be used for the pure weighted scheme, or the pure hierarchical scheme, or the hybrid combination. These approaches are verified and implemented on our virtual quadruped-on-wheel robot with one manipulator on its floating base. The results show that the control schemes with the prioritized impedance controller enable our robot to handle complex missions efficiently and compliantly.

Regarding the motion generators for the quadruped-on-wheel robot, *TowrISIR*, we apply the extra centroidal properties of four-limb robots in addition to the generalized dynamics models. We propose one wheel-motion generator and one whole-body motion generator to track the robot centroidal motion reference. All wheels are in contact with the ground, and the legged motion is used as an active suspension and the base height control. In the first wheel motion generator, we assume that the legged suspension is already known, and then the wheel motion can be achieved by combining the whole-body inverse kinematics model and the centroidal momentum & dynamics model. In the second whole-body motion generator, we develop one altitude controller by controlling the relative motion between the base and the robot CoM, which enables the robot to run on terrains with much-difference altitude. Then the whole-body motion generator is developed by combining both the kinematics model, the wheel-center motion model, the robot centroidal momentum and dynamics models, and the attitude control model. Finally, the robot is simulated to be driven on various rough terrains with compatible dynamic behaviors. This work serves as the basis to achieve the hybrid locomotion modes in the future work by combining the dynamic-legged motion and the wheeled motion together.

For the quadruped robot, *QuadISIR*, we propose one new locomotion principle and one trotting-mode motion generator without the leg-massless assumption. This new principle further closes the gap between

the quadruped robots and their counterparts. For the trotting locomotion gait, the whole-body CoM or the base motion is defined initially, then the stance legs follow the base motion, and the swing legs compensate the delay influence from the stance-leg motion to the centroidal motion. This new principle is more consistent with the locomotion logic of the quadruped animals. We propose the more general legged internal relationships, which can be applied for forward and lateral trotting and turn trotting motions. This legged motion generator also exploits the strengths of the centroidal dynamics of the whole robot and each leg. The developed motion generator can also be applied for the walking mode which is applied when two swing legs land in different timings. Instead of using the leg real joints, the virtual joints between the floating base and each leg CoM are applied to build the robot mass-distribution model, centroidal momentum and dynamics models. By combining the whole-body kinematics model and the kinematics constraints relating to the contact constraints and the parallel mechanism, the legged motion generator enables the robot to trot and walk without pre-defined legged trajectories. In addition, the relative motion between the robot CoM and the base is controlled instead of their absolute motions, which enables the robot to recover from external push. We propose the new state machine for switching the locomotion gaits between walking and trotting, and this state machine also serves the torque controller which determines the dimension of the legged control states and the contact Jacobian. The validation is performed through advanced numerical simulations using our PIC based torque controller. Our new locomotion principle integrates the system design, motion generation and whole-body torque control together, and this common one framework is able to handle several locomotion gaits. The robot succeeds to trot in the forward and lateral directions, turn, handle push-recovery motion, as well as trot on the relative rough terrain.

## 6.2 Perspectives

### **Wheeled Locomotion:**

For the rolling-mode locomotion, we will develop a more accurate estimation method of the wheel contact points. This allows for a more accurate Equation of Motion and benefits the model based torque controller and motion generators. The desired centroidal angular motion is generated on-line. However, in our opinion, the angular property can influence the whole-robot stability [Lee 2012]. Therefore, we will develop an angular momentum based general stability criterion.

### **Legged Locomotion:**

For the legged locomotion mode, although the principle in Chapter 5 is the start of generating the legged motion on-line using the robot dynamics, this work opens a door leading to robots being able to rely only on their own properties to run, without pre-defined legged trajectories. Future works will focus on the following:

- The developed legged motion generator will be applied to the amble gait.
- For the fast locomotion modes, e.g. trotting, bounding and galloping with flight phases, we believe that the passive dynamics properties can enable the automatic and energy-efficient motion generation without pre-defined leg trajectories.
- In our opinion, the whole-body compliance comes from both the kinematics and dynamics (torque control) levels. In our legged motion generator, only one method relating to the vertical motion between the base and the CoM is applied, however, it induces discontinuity in the position and velocity levels. The method in [Barasuol 2013] brings a potential approach to handle the compliance at the kinematics level. Therefore, we may borrow this idea and develop a better kinematics compliance method.
- In our developed motion generator, we define the base and the CoM to share the similar motions in the forward and lateral directions for two locomotion modes. However, they can be set to be different to achieve various locomotion gaits.

- In the legged motion generator, we achieve the internal relationships of the swing legs by using the stance-leg motion directions. This seems to be not natural for more complex gaits. Therefore, we will focus on how to set a whole-body angular motion reference corresponding to the environment. In this way, the relationship between the swing legs are not necessary.
- Similarly with the motion generation for Tower *ISIR*, we will find an angular motion relationship between the base and the centroidal motions to develop a general stability criterion for legged robots. The control of the angular motion references are supposed to be even closer to the locomotion principle of quadruped animals.
- In the state of the art, there is still no practical methods to achieve the centroidal orientation. Therefore, we can develop an approach to estimate the unapparent centroidal orientation angles by comparing the running state with the robot normal state. This could allow the desired swing-leg positions to be expressed using only the robot mass distribution model and the centroidal orientation model, which will be independent of the legged internal relationship.
- The robot structure can be improved to provide the dynamics model with more simple linear properties, especially between the actuated torque and the legged motion. This can be more advantageous for the whole-body motion optimization with actuated-torque constraints.

In the long-work plan, we will combine the rolling and legged motion together, which can provide the robot with more abilities to handle complex and changing environment, e.g. trotting-rolling, jumping-rolling locomotion modes. The changing wheel-contact points induces influences to the contact constraints of the legged motion. We will focus our research on developing a general motion generator without defining the robot internal motion trajectories.

# Robot Structure

In this appendix, we give more descriptions of our four-limb robots. Since that the floating bases of the two robots share the similar design structures, we only introduce one detailed base layout and give the differences of the leg structures. In Fig. A.1, Fig. A.2 and Fig. A.3, we show the designed whole-body structure. Then in Fig. A.4, the whole-robot electrical and electric control layouts are depicted, including the network protocols, recognition sensors, power supplies, control system, one leg electrical structure, especially the detailed connection relationships. The power source is from two electric batteries which drive the pump motor and four wheel motors, and supply electricity to the controller, motor drivers and the sensors. In one leg, one hydraulic rotary single-vane actuator is selected for the HAA joint, one hydraulic double-vane actuator for the HFE joint, one hydraulic linear actuator for the KFE joint, one high-torque electric motor plus one planetary reducer for the wheel joint. Each hydraulic actuator is driven by a high-bandwidth servo valve which controls the fluid pressure and volume from a high working pressure. The control field bus applies the EtherCAT protocol to achieve the high control rate. The sensors include joint torque sensors, load cells, absolute and incremental encoders, IMU, GPS, Lidar and camera, which give back the environment information and robot configurations. The mechanical design integrates the power, controller, actuation, perception, safety into the limited space and allows the robot abilities to run on rough terrain with highly dynamics. Then the designed 6-DOF manipulator structure is depicted in Fig. A.5, which is assembled on the floating base of *TowrISIR* to achieve more complex grasp tasks, detailed in Chapter 3. The leg structure of *TowrISIR* is shown in Fig. A.6. Then we show the leg-structure difference of *QuadISIR* in Fig. A.7, which applies the parallelogram mechanism. We apply the similar selection principle for the hydraulic actuation and control system as that in HyQ series robots in [Semini 2010] and [Semini 2017], even though the different robot-structure layout is applied.

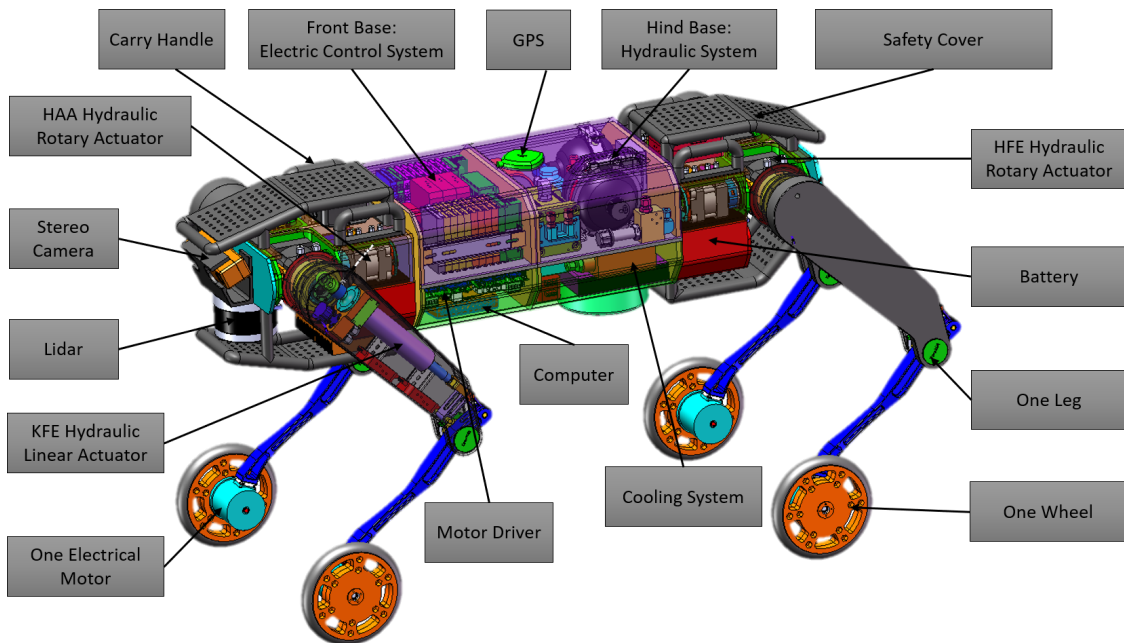


Figure A.1: Whole-body structure of *TowrISIR*.

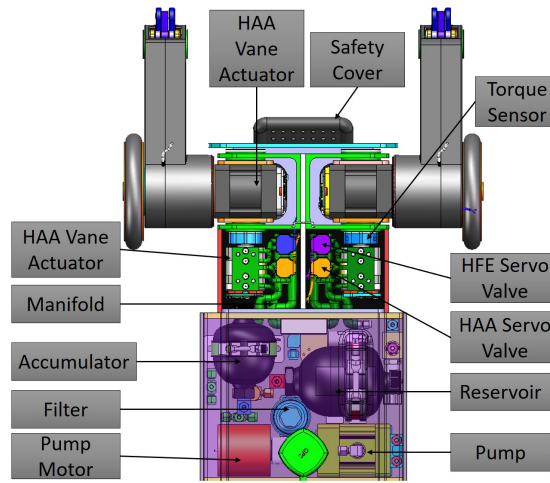


Figure A.2: Hind half structure of TowerISIR shows the hydraulic control components.

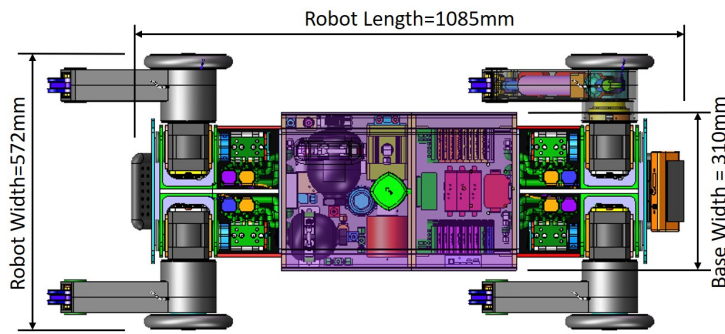


Figure A.3: Floating base layout and dimension of TowerISIR.

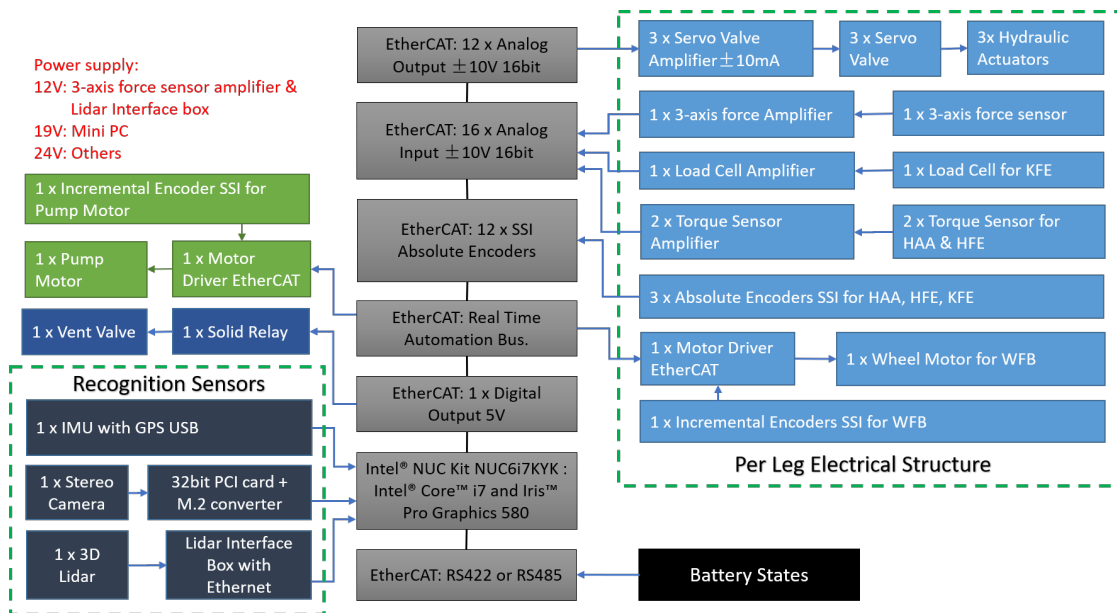


Figure A.4: Electric Architecture of TowerISIR.

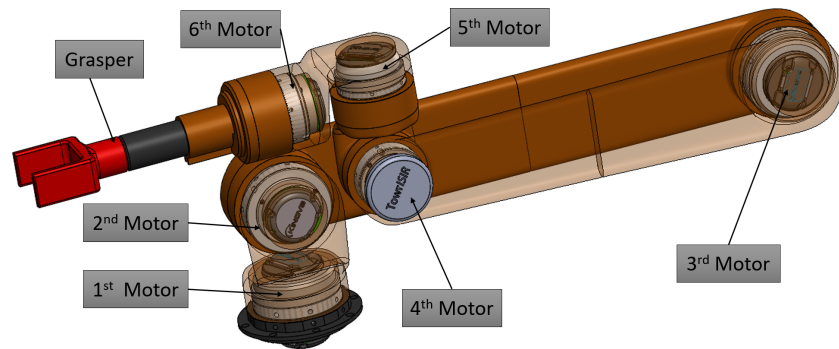


Figure A.5: 6-DOF manipulator structure for TowrISIR.

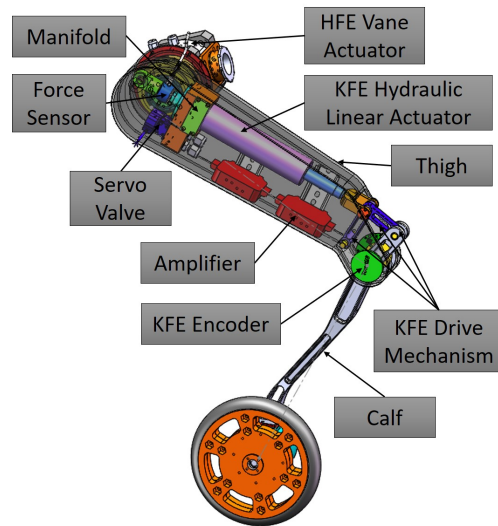


Figure A.6: One leg structure of TowrISIR.

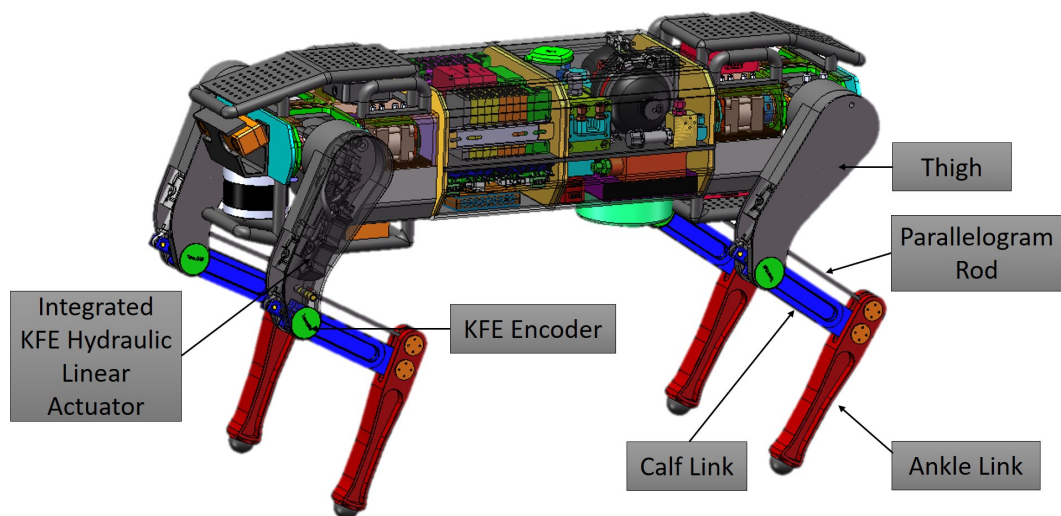


Figure A.7: One leg structure of QuadISIR with parallelogram mechanism.



# Simulation Environment

Since we use simulations to verify the methods in this thesis, therefore, the simulated robots are applied to be as same as the designed 3D model in the CAD software. The simulated robots should have enough details to be as real as the CAD models, which is beneficial to imitate the experiments. In this appendix, we will introduce the establishment process of the simulation scenarios, which is similar for each simulation environment.

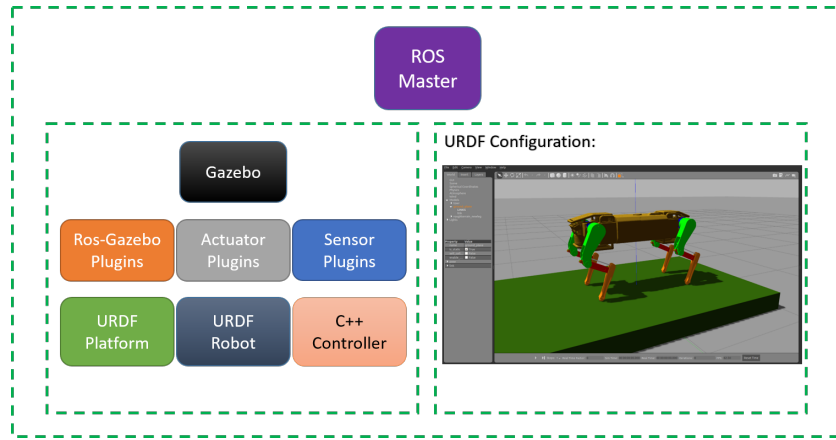


Figure B.1: The components required by the simulation environment.

Initially, we build the 3D robot models in Solidworks (Education Edition), as shown in Fig. 2.1 and Fig. 5.1. It is exported as the URDF file which is required by the ROS-GAZEBO simulation engine. This file should be modified to integrate the link modules, joint modules, transmission modules, sensor modules and the environment settings together. It is noticed that the electrical and hydraulic components inside the base can be deleted to enable the engine with faster speed. In addition to the robot, some artificial platforms should be built in Solidworks and then are also exported as URDF files. These files will be launched by ROS and generated in GAZEBO environment and we apply the default collision and friction setting as the GAZEBO configuration. In addition, the C++ language code relates to the system modeling, motion generators and torque controllers developed in this thesis, which controls the robot in GAZEBO through the ROS node. The simulation environment is shown in Fig. B.1 and we have the following settings:

- We configure the collision links in URDF files to be less complex than the visual links, which enables the collision detection to be faster in GAZEBO environment.
- Parallelogram mechanisms cannot be configured in the URDF files. We use the closed-loop plugin developed in [github.com/wojiaojiao/pegasus\\_gazebo\\_plugins](https://github.com/wojiaojiao/pegasus_gazebo_plugins) to create the parallelogram mechanisms for the legs of QuadISIR.
- The self-collision is turned off among the bodies of one robot. The max-contact point number is set to be 1 for the wheels and feet of TowrISIR and QuadISIR.
- The stiffness and damping of each collision body and the platforms are set with the default values,  $k_p = 1000000$  and  $k_d = 10$ . The Coulomb friction coefficients in two directions are also set by default with big values,  $\mu_1 = \mu_2 = 90000$ .

# List of Publications

## Papers as First Author

1. Wenqian Du, Faïz Benamar. *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). doi: 10.1007/s11044-020-09728-y.
2. Wenqian Du, Mohamed Fnadi, Faïz Benamar. *Whole-Body Motion Tracking for a Quadruped-on-Wheel Robot via a Compact-Form Controller With Improved Prioritized Optimization*. *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 516-523, April 2020. doi: 10.1109/LRA.2019.2963822.
3. Wenqian Du, Mohamed Fnadi, Faïz Benamar. *Rolling based Locomotion on Rough Terrain for a Wheeled Quadruped using Centroidal Dynamics*. *Mechanism and Machine Theory*, vol 153, 103984, doi: 10.1016/j.mechmachtheory.2020.103984.
4. Wenqian Du, Mohamed Fnadi, Faïz Benamar. *Integration of Prioritized Impedance Controller in Hierarchical Operational-Space Torque Control Frameworks for Locomotion Robots*. *Robotics and Autonomous Systems*, 2020. (Under Review)
5. Wenqian Du, Mohamed Fnadi, Faïz Benamar. *A New Whole-body Motion Generator and Adaptive Attitude Control for a Quadruped-on-Wheel Robot Rolling on Rough Terrain*. *Autonomous Robots*, 2020. (Under Review)
6. Wenqian Du, Mohamed Fnadi, Etienne Moullet, Faïz Benamar. *Leg Centroidal Dynamics based New Locomotion Principle of Quadruped Robots with On-line Legged Motion Generation*. *Autonomous Robots*, 2020. (Under Review)
7. Wenqian Du, Ze Wang, Etienne Moullet, Faïz Benamar. *Meaningful Centroidal Frame Orientation of Multi-body Floating Locomotion Systems*. *IEEE Robotics and Automation Letters*, 2020. (Under Review)

## Papers as Second Author

1. Mohamed Fnadi, Wenqian Du, Frédéric Plumet, Faïz Benamar. *Local Obstacle-skirting Path Planning for a Fast Bi-steerable Rover using Béziere Curves*. 2020 International Conference on Robotics and Automation (ICRA) Paris. Pages 1–9.
2. Mohamed Fnadi, Wenqian Du, Frédéric Plumet, Faïz Benamar. *Constrained Model Predictive Control based Dynamic Path Tracking of a Bi-steerable Rover on Slippery Grounds*. *Control Engineering Practice*, vol 107, 103984, 2021, doi.org/10.1016/j.conengprac.2020.104693.

# Bibliography

- [Aceituno-Cabezas 2017] Bernardo Aceituno-Cabezas, Carlos Mastalli, Hongkai Dai, Michele Focchi, Andreea Radulescu, Darwin G Caldwell, José Cappelletto, Juan C Grieco, Gerardo Fernández-López and Claudio Semini. *Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization*. IEEE Robotics and Automation Letters, vol. 3, no. 3, pages 2531–2538, 2017. (Cited on page 10.)
- [Albu-Schaffer 2002] Alin Albu-Schaffer and Gerd Hirzinger. *Cartesian impedance control techniques for torque controlled light-weight robots*. In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, volume 1, pages 657–663. IEEE, 2002. (Cited on page 16.)
- [Albu-Schäffer 2007] Alin Albu-Schäffer, Christian Ott and Gerd Hirzinger. *A unified passivity-based control framework for position, torque and impedance control of flexible joint robots*. The international journal of robotics research, vol. 26, no. 1, pages 23–39, 2007. (Cited on page 17.)
- [An 2012] Sang-ik An, Yonghwan Oh and Dong-Soo Kwon. *Zero-moment point based balance control of leg-wheel hybrid structures with inequality constraints of kinodynamic behavior*. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2471–2477. IEEE, 2012. (Cited on page 11.)
- [Andrade 1998] G Andrade, F BenAmar, Ph Bidaud and R Chatila. *Modeling wheel-sand interaction for optimization of a rolling-peristaltic motion of a marsokhod robot*. In International Conference on Intelligent Robots and Systems, pages 576–581, 1998. (Cited on page 6.)
- [Arslan 2009] Omur Arslan, Uluc Saranlı and Omer Morgul. *An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions*. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 2388–2393. IEEE, 2009. (Cited on page 8.)
- [Banala 2009] Sai K Banala, Seok Hun Kim, Sunil K Agrawal and John P Scholz. *Robot assisted gait training with active leg exoskeleton (ALEX)*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 17, no. 1, pages 2–8, 2009. (Cited on page 17.)
- [Barasuol 2011] Victor Barasuol, Victor Julianano De Negri and Edson Roberto De Pieri. *WCPG: a central pattern generator for legged robots based on workspace intentions*. In Proceedings of the ASME dynamic system and control conference (DSCC), pages 111–114, 2011. (Cited on pages xiii and 9.)
- [Barasuol 2013] Victor Barasuol, Jonas Buchli, Claudio Semini, Marco Frigerio, Edson R De Pieri and Darwin G Caldwell. *A reactive controller framework for quadrupedal locomotion on challenging terrain*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 2554–2561. IEEE, 2013. (Cited on pages 9 and 113.)
- [Bellicoso 2016] C Dario Bellicoso, Christian Gehring, Jemin Hwangbo, Péter Fankhauser and Marco Hutter. *Perception-less terrain adaptation through whole body control and hierarchical optimization*. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 558–564. IEEE, 2016. (Cited on pages 15 and 46.)
- [Besseron 2008] Guillaume Besseron, Ch Grand, F Ben Amar and Ph Bidaud. *Decoupled control of the high mobility robot hylas based on a dynamic stability margin*. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 2435–2440. IEEE, 2008. (Cited on page 10.)
- [Bingquan 2014] Shen Bingquan. *Design and control methodology of a lower extremity assistive device*. PhD thesis, 2014. (Cited on page 17.)

- [Bjelonic 2020] Marko Bjelonic, Prajish K Sankar, C Dario Bellicoso, Heike Vallery and Marco Hutter. *Rolling in the Deep—Hybrid Locomotion for Wheeled-Legged Robots using Online Trajectory Optimization*. IEEE Robotics and Automation Letters, vol. 5, no. 2, pages 3626–3633, 2020. (Cited on page 7.)
- [Blickhan 1989] Reinhard Blickhan. *The spring-mass model for running and hopping*. Journal of biomechanics, vol. 22, no. 11-12, pages 1217–1227, 1989. (Cited on page 8.)
- [Blickhan 1993] Reinhard Blickhan and RJ Full. *Similarity in multilegged locomotion: bouncing like a monopode*. Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology, vol. 173, no. 5, pages 509–517, 1993. (Cited on page 8.)
- [Boaventura 2012] Thiago Boaventura, Claudio Semini, Jonas Buchli, Marco Frigerio, Michele Focchi and Darwin G Caldwell. *Dynamic torque control of a hydraulic quadruped robot*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1889–1894. IEEE, 2012. (Cited on page 16.)
- [Boaventura 2013a] Thiago Boaventura. *Hydraulic compliance control of the quadruped robot HyQ*. Doctor of Philosophy PhD Thesis, Advanced Robotics Department, University of Genova, Genova, Italy, 2013. (Cited on page 16.)
- [Boaventura 2013b] Thiago Boaventura, Gustavo A Medrano-Cerda, Claudio Semini, Jonas Buchli and Darwin G Caldwell. *Stability and performance of the compliance controller of the quadruped robot HyQ*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 1458–1464. IEEE, 2013. (Cited on page 16.)
- [Boaventura 2015] Thiago Boaventura, Jonas Buchli, Claudio Semini and Darwin G Caldwell. *Model-based hydraulic impedance control for dynamic robots*. IEEE Transactions on Robotics, vol. 31, no. 6, pages 1324–1336, 2015. (Cited on page 17.)
- [Bouloubasis 2007] Antonios K Bouloubasis and Gerard T McKee. *The mobility system of the multi-tasking rover (MTR)*. In Robotics and Automation, 2007 IEEE International Conference on, pages 4919–4924. IEEE, 2007. (Cited on page 6.)
- [Bouton 2012] A Bouton, Ch Grand, F Ben Amar, F Ben Amar, Ch Grand, A Bouton, Ch Grand, F Ben Amar, JM Jehanno, A Cully *et al.* *Design and Control of a Compliant Wheel-on-Leg Rover which Conforms to Uneven Terrain*. Transaction on Mechatronics, vol. 83, no. 2012, pages 1177–1184, 2012. (Cited on page 11.)
- [Bouton 2016] Arthur Bouton, Christophe Grand and Faïz Benamar. *Motion control of a compliant wheel-leg robot for rough terrain crossing*. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, pages 2846–2851. IEEE, 2016. (Cited on page 7.)
- [BOUTON 2017a] ARTHUR BOUTON, FAÏZ BENAMAR and CHRISTOPHE GRAND. *DESIGN OF A COMPLIANT WHEEL-LEGGED ROBOT*. In Advances in Cooperative Robotics, pages 498–505. World Scientific, 2017. (Cited on pages 6 and 7.)
- [Bouton 2017b] Arthur Bouton, Christophe Grand and Faïz Benamar. *Obstacle negotiation learning for a compliant wheel-on-leg robot*. In Robotics and Automation (ICRA), 2017 IEEE International Conference on, pages 2420–2425. IEEE, 2017. (Cited on page 7.)
- [Bouyarmane 2011] Karim Bouyarmane and Abderrahmane Kheddar. *Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4414–4419. IEEE, 2011. (Cited on pages 15 and 41.)

- [Buchli 2007] Jonas Buchli. *Engineering limit cycle systems: Adaptive frequency oscillators and applications to adaptive locomotion control of compliant robots*. In Ph. D. dissertation, Inst. Bioeng., Swiss Fed. Inst. Technol., Lausanne, Switzerland, 2007. (Cited on page 9.)
- [Buchli 2009] Jonas Buchli, Mrinal Kalakrishnan, Michael Mistry, Peter Pastor and Stefan Schaal. *Compliant quadruped locomotion over rough terrain*. In Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, pages 814–820. IEEE, 2009. (Cited on page 17.)
- [Buehler 2005] M Buehler, R Playter and M Raibert. *Robots step outside*. In Int. Symp. Adaptive Motion of Animals and Machines (AMAM), Ilmenau, Germany, pages 1–4, 2005. (Cited on pages 3 and 4.)
- [Cao 2014] Jinghui Cao, Sheng Quan Xie, Raj Das and Guo L Zhu. *Control strategies for effective robot assisted gait rehabilitation: the state of art and future prospects*. Medical engineering & physics, vol. 36, no. 12, pages 1555–1566, 2014. (Cited on page 17.)
- [Caputo 2015] Joshua M Caputo. *Informing ankle-foot prosthesis design and prescription through systematic experimentation with a tethered robotic prosthesis*. PhD thesis, Carnegie Mellon University, 2015. (Cited on page 17.)
- [Caron 2015] Stéphane Caron, Quang-Cuong Pham and Yoshihiko Nakamura. *Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics*. In Robotics: Science and Systems, 2015. (Cited on page 11.)
- [Dafarra 2016] Stefano Dafarra, Francesco Romano and Francesco Nori. *Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub humanoid robot*. In Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, pages 152–157. IEEE, 2016. (Cited on page 13.)
- [Dai 2014] Hongkai Dai, Andrés Valenzuela and Russ Tedrake. *Whole-body motion planning with centroidal dynamics and full kinematics*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 295–302. IEEE, 2014. (Cited on pages 10 and 15.)
- [Dallali 2011] Houman Dallali. *Modelling and dynamic stabilisation of a compliant humanoid robot, CoMan*. PhD thesis, 2011. (Cited on page 17.)
- [De Lasa 2010] Martin De Lasa, Igor Mordatch and Aaron Hertzmann. *Feature-based locomotion controllers*. In ACM Transactions on Graphics (TOG), volume 29, page 131. ACM, 2010. (Cited on pages 15, 46 and 48.)
- [De Luca 2006] Alessandro De Luca, Giuseppe Oriolo and Paolo Robuffo Giordano. *Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 1867–1873. IEEE, 2006. (Cited on page 10.)
- [De Luca 2010] Alessandro De Luca, Giuseppe Oriolo and Paolo Robuffo Giordano. *Kinematic control of nonholonomic mobile manipulators in the presence of steering wheels*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 1792–1798. IEEE, 2010. (Cited on page 10.)
- [Deits 2014] Robin Deits and Russ Tedrake. *Footstep planning on uneven terrain with mixed-integer convex optimization*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 279–286. IEEE, 2014. (Cited on pages 10 and 15.)
- [Dietrich 2011a] Alexander Dietrich, Thomas Wimböck and Alin Albu-Schäffer. *Dynamic whole-body mobile manipulation with a torque controlled humanoid robot via impedance control laws*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 3199–3206. IEEE, 2011. (Cited on page 17.)

- [Dietrich 2011b] Alexander Dietrich, Thomas Wimböck, Alin Albu-Schäffer and Gerd Hirzinger. *Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 6136–6142. IEEE, 2011. (Cited on page 17.)
- [Dietrich 2012] Alexander Dietrich, Thomas Wimbock, Alin Albu-Schaffer and Gerd Hirzinger. *Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom*. IEEE Robotics & Automation Magazine, vol. 19, no. 2, pages 20–33, 2012. (Cited on pages 12, 15, 17, 40 and 43.)
- [Dietrich 2015] Alexander Dietrich, Kristin Bussmann, Florian Petit, Paul Kotyczka, Christian Ott, Boris Lohmann and Alin Albu-Schäffer. *Whole-body impedance control of wheeled mobile manipulators: Stability analysis and experiments on the humanoid robot Rollin’Justin*. Autonomous Robots, 2015. (Cited on page 5.)
- [Dietrich 2016] Alexander Dietrich. *Whole-body impedance control of wheeled humanoid robots*, volume 116. Springer, 2016. (Cited on pages 5, 6 and 17.)
- [Doha 2011] Eid H Doha, AH Bhrawy and MA Saker. *Integrals of Bernstein polynomials: an application for the solution of high even-order differential equations*. Applied Mathematics Letters, vol. 24, no. 4, pages 559–565, 2011. (Cited on page 88.)
- [Du 2020a] Wenqian Du and Faiz Benamar. *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, pages 1–17, 2020. (Cited on pages 18, 35 and 41.)
- [Du 2020b] Wenqian Du, Mohamed Fnadi and Faiz Benamar. *Leg centroidal dynamics based new locomotion principle of quadruped robots with on-line legged motion generation*. International Journal of Robotics Research (Under Review), 2020. (Cited on page 19.)
- [Du 2020c] Wenqian Du, Mohamed Fnadi and Faiz Benamar. *A new whole-body motion generator and adaptive altitude control for a quadruped-on-wheel robot*. Autonomous Robots (Under Review), 2020. (Cited on page 18.)
- [Du 2020d] Wenqian Du, Mohamed Fnadi and Faiz Benamar. *Prioritized impedance controller in hierarchical operational-space control framework for legged robots*. Robotics and Autonomous Systems (Under Review), 2020. (Cited on page 18.)
- [Du 2020e] Wenqian Du, Mohamed Fnadi and Faiz Benamar. *Rolling based locomotion on rough terrain for a wheeled quadruped using centroidal dynamics*. Mechanism and Machine Theory, vol. 153, page 103984, 2020. (Cited on page 18.)
- [Du 2020f] Wenqian Du, Mohamed Fnadi and Faiz Benamar. *Whole-Body Motion Tracking for a Quadruped-on-Wheel Robot via a Compact-Form Controller With Improved Prioritized Optimization*. IEEE Robotics and Automation Letters, vol. 5, no. 2, pages 516–523, 2020. (Cited on pages 18, 42 and 98.)
- [Duschau-Wicke 2010] Alexander Duschau-Wicke, Joachim von Zitzewitz, Andrea Caprez, Lars Lunenburger and Robert Riener. *Path control: a method for patient-cooperative robot-aided gait rehabilitation*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 18, no. 1, pages 38–48, 2010. (Cited on page 17.)
- [Ekkelenkamp 2005] R Ekkelenkamp, J Veneman and Herman van der Kooij. *LOPES: Selective control of gait functions during the gait rehabilitation of CVA patients*. In Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on, pages 361–364. IEEE, 2005. (Cited on page 17.)



- [Englsberger 2014] Johannes Engelsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid *et al.* *Overview of the torque-controlled humanoid robot TORO*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 916–923. IEEE, 2014. (Cited on page 17.)
- [Ernst 2010] Michael Ernst, Hartmut Geyer and Reinhard Blickhan. *Spring-legged locomotion on uneven ground: a control approach to keep the running speed constant*. In Mobile Robotics: Solutions and Challenges, pages 639–644. World Scientific, 2010. (Cited on page 8.)
- [Escande 2014] Adrien Escande, Nicolas Mansard and Pierre-Brice Wieber. *Hierarchical quadratic programming: Fast online humanoid-robot motion generation*. The International Journal of Robotics Research, vol. 33, no. 7, pages 1006–1028, 2014. (Cited on pages 15, 17 and 43.)
- [Featherstone 1983] Roy Featherstone. *The calculation of robot dynamics using articulated-body inertias*. The International Journal of Robotics Research, vol. 2, no. 1, pages 13–30, 1983. (Cited on page 13.)
- [Featherstone 2014] Roy Featherstone. Rigid body dynamics algorithms. Springer, 2014. (Cited on pages 13, 21, 22, 24, 27, 29, 76 and 82.)
- [Feng 2016] Siyuan Feng. *Online hierarchical optimization for humanoid control*. 2016. (Cited on page 15.)
- [Fernández 2010] Gea Fernández *et al.* *Predictive context-based adaptive compliance for interaction control of robot manipulators*. PhD thesis, Bremen, Univ., Diss., 2011, 2010. (Cited on page 16.)
- [Ferreau 2014] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock and Moritz Diehl. *qpOASES: A parametric active-set algorithm for quadratic programming*. Mathematical Programming Computation, vol. 6, no. 4, pages 327–363, 2014. (Cited on pages 46 and 48.)
- [Focchi 2013] Michele Focchi. *Strategies to improve the impedance control performance of a quadruped robot*. PhD thesis, Ph. D thesis, Istituto Italiano di Tecnologia, Genoa, Italy, 2013. (Cited on page 16.)
- [Focchi 2016] Michele Focchi, Gustavo A Medrano-Cerda, Thiago Boaventura, Marco Frigerio, Claudio Semini, Jonas Buchli and Darwin G Caldwell. *Robot impedance control and passivity analysis with inner torque and velocity feedback loops*. Control Theory and Technology, vol. 14, no. 2, pages 97–112, 2016. (Cited on page 17.)
- [Focchi 2017] Michele Focchi, Andrea Del Prete, Ioannis Havoutis, Roy Featherstone, Darwin G Caldwell and Claudio Semini. *High-slope terrain locomotion for torque-controlled quadruped robots*. Autonomous Robots, vol. 41, no. 1, pages 259–272, 2017. (Cited on page 15.)
- [Garofalo 2015] Gianluca Garofalo, Bernd Henze, Johannes Engelsberger and Christian Ott. *On the inertially decoupled structure of the floating base robot dynamics*. IFAC-PapersOnLine, vol. 48, no. 1, pages 322–327, 2015. (Cited on page 13.)
- [Gehring 2013] Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A Hoepflinger and Roland Siegwart. *Control of dynamic gaits for a quadrupedal robot*. In Robotics and automation (ICRA), 2013 IEEE international conference on, pages 3287–3292. IEEE, 2013. (Cited on page 8.)
- [Gehring 2015] Christian Gehring, C Dario Bellicoso, Stelian Coros, Michael Bloesch, Péter Fankhauser, Marco Hutter and Roland Siegwart. *Dynamic trotting on slopes for quadrupedal robots*. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5129–5135. IEEE, 2015. (Cited on page 91.)



- [Geyer 2005] Hartmut Geyer, Andre Seyfarth and Reinhard Blickhan. *Spring-mass running: simple approximate solution and application to gait stability*. Journal of theoretical biology, vol. 232, no. 3, pages 315–328, 2005. (Cited on page 8.)
- [Geyer 2006] Hartmut Geyer, Andre Seyfarth and Reinhard Blickhan. *Compliant leg behaviour explains basic dynamics of walking and running*. Proceedings of the Royal Society of London B: Biological Sciences, vol. 273, no. 1603, pages 2861–2867, 2006. (Cited on page 8.)
- [Giordano 2009] P Robuffo Giordano, Matthias Fuchs, A Albu-Schaffer and Gerd Hirzinger. *On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs*. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pages 4080–4087. IEEE, 2009. (Cited on page 10.)
- [Gong 2016] Chen Gong. *Development of a robotic exoskeleton system for gait rehabilitation*. PhD thesis, National University of Singapore (Singapore), 2016. (Cited on page 17.)
- [Gong 2019] Yukai Gong, Ross Hartley, Xingye Da, Ayonga Hereid, Omar Harib, Jiunn-Kai Huang and Jessy Grizzle. *Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway*. In 2019 American Control Conference (ACC), pages 4559–4566. IEEE, 2019. (Cited on page 78.)
- [Goswami 2004] Ambarish Goswami and Vinutha Kalleem. *Rate of change of angular momentum and balance maintenance of biped robots*. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, volume 4, pages 3785–3790. IEEE, 2004. (Cited on page 13.)
- [Grand 2004] Christophe Grand, Faiz Benamar, Frédéric Plumet and Philippe Bidaud. *Stability and traction optimization of a reconfigurable wheel-legged robot*. The International Journal of Robotics Research, vol. 23, no. 10-11, pages 1041–1058, 2004. (Cited on pages 10 and 11.)
- [Grand 2010] Christophe Grand, Faiz Benamar and Frédéric Plumet. *Motion kinematics analysis of wheeled-legged rover over 3D surface with posture adaptation*. Mechanism and Machine Theory, vol. 45, no. 3, pages 477–495, 2010. (Cited on pages xiii, 6, 7, 10 and 11.)
- [Grand 2016] Christophe Grand, Pierre Jarrault, Faiz Ben Amar and Philippe Bidaud. *Experimental evaluation of obstacle clearance by a hybrid wheel-legged robot*. In Experimental Robotics, pages 47–58. Springer, 2016. (Cited on pages 6, 7 and 10.)
- [Grotzinger 2012] John P Grotzinger, Joy Crisp, Ashwin R Vasavada, Robert C Anderson, Charles J Baker, Robert Barry, David F Blake, Pamela Conrad, Kenneth S Edgett, Bobak Ferdowski *et al*. *Mars Science Laboratory mission and science investigation*. Space science reviews, vol. 170, no. 1-4, pages 5–56, 2012. (Cited on page 6.)
- [Han 2014] Bin Han, Xin Luo, Qingyu Liu, Bo Zhou and Xuedong Chen. *Hybrid control for SLIP-based robots running on unknown rough terrain*. Robotica, vol. 32, no. 7, pages 1065–1080, 2014. (Cited on page 8.)
- [Hauser 2008] Kris Hauser, Timothy Bretl, Jean-Claude Latombe and Brian Wilcox. *Motion planning for a six-legged lunar robot*. In Algorithmic Foundation of Robotics VII, pages 301–316. Springer, 2008. (Cited on page 6.)
- [Herr 2008] Hugh Herr and Marko Popovic. *Angular momentum in human walking*. Journal of Experimental Biology, vol. 211, no. 4, pages 467–481, 2008. (Cited on page 13.)
- [Herzog 2013] Alexander Herzog, Ludovic Righetti, Felix Grimmering, Peter Pastor and Stefan Schaal. *Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid*. arXiv preprint, 2013. (Cited on page 15.)

- [Herzog 2014] Alexander Herzog, Ludovic Righetti, Felix Grimmering, Peter Pastor and Stefan Schaal. *Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics*. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 981–988. IEEE, 2014. (Cited on pages 14 and 15.)
- [Herzog 2016] Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimmering, Stefan Schaal and Ludovic Righetti. *Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid*. Autonomous Robots, vol. 40, no. 3, pages 473–491, 2016. (Cited on pages 14, 15, 42 and 46.)
- [Heverly 2010] Matt Heverly, Jaret Matthews, Matt Frost and Chris Quin. *Development of the Tri-ATHLETE lunar vehicle prototype*. 2010. (Cited on pages 6 and 7.)
- [Hogan 1984] Neville Hogan. *Impedance control: An approach to manipulation*. In American Control Conference, 1984, pages 304–313. IEEE, 1984. (Cited on page 16.)
- [Hopkins 2015a] Michael A Hopkins, Dennis W Hong and Alexander Leonessa. *Compliant locomotion using whole-body control and Divergent Component of Motion tracking*. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 5726–5733. IEEE, 2015. (Cited on pages 15 and 17.)
- [Hopkins 2015b] Michael Anthony Hopkins. *Dynamic Locomotion and Whole-Body Control for Compliant Humanoids*. 2015. (Cited on page 15.)
- [Hoyt 1981] Donald F Hoyt and C Richard Taylor. *Gait and the energetics of locomotion in horses*. Nature, vol. 292, no. 5820, pages 239–240, 1981. (Cited on page 16.)
- [Hutter 2010] Marco Hutter, C David Remy, Mark A Höpflinger and Roland Siegwart. *Slip running with an articulated robotic leg*. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 4934–4939. IEEE, 2010. (Cited on page 8.)
- [Hutter 2011] Marco Hutter, C David Remy, Mark A Hoepflinger and Roland Siegwart. *High compliant series elastic actuation for the robotic leg ScarLETH*. In Proc. of the International Conference on Climbing and Walking Robots (CLAWAR), numéro EPFL-CONF-175826, 2011. (Cited on page 4.)
- [Hutter 2012] Marco Hutter, Michael Gehring, Michael Bloesch, C David Remy, Roland Yves Siegwart and Mark A Hoepflinger. *StarLETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion*. 2012. (Cited on page 17.)
- [Hutter 2013a] Marco Hutter. *StarLETH & Co-design and control of legged robots with compliant actuation*. PhD thesis, 2013. (Cited on pages 1, 13 and 15.)
- [Hutter 2013b] Marco Hutter, Mark A Hoepflinger, Christian Gehring, Michael Bloesch, C David Remy and Roland Siegwart. *Hybrid operational space control for compliant legged systems*. Robotics, page 129, 2013. (Cited on page 12.)
- [Hutter 2014] Marco Hutter, Hannes Sommer, Christian Gehring, Mark Hoepflinger, Michael Bloesch and Roland Siegwart. *Quadrupedal locomotion using hierarchical operational space control*. The International Journal of Robotics Research, vol. 33, no. 8, pages 1047–1062, 2014. (Cited on pages 8, 11 and 15.)
- [Hutter 2016] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C Dario Bellicoso, Vasiliios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch *et al.* *Anymal-a highly mobile and dynamic quadrupedal robot*. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pages 38–44. IEEE, 2016. (Cited on pages 4 and 17.)

- [Hyun 2014] Dong Jin Hyun, Sangok Seok, Jongwoo Lee and Sangbae Kim. *High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah*. The International Journal of Robotics Research, vol. 33, no. 11, pages 1417–1445, 2014. (Cited on pages 9, 78 and 88.)
- [Iagnemma 2003] Karl Iagnemma, Adam Rzepniewski, Steven Dubowsky and Paul Schenker. *Control of robotic vehicles with actively articulated suspensions in rough terrain*. Autonomous Robots, vol. 14, no. 1, pages 5–16, 2003. (Cited on page 6.)
- [Ijspeert 2008] Auke Jan Ijspeert. *Central pattern generators for locomotion control in animals and robots: a review*. Neural networks, vol. 21, no. 4, pages 642–653, 2008. (Cited on page 9.)
- [Jinfu 2015] Li Jinfu. *Assistive Device for Lower Extremity Gait Training and Assistance*. PhD thesis, 2015. (Cited on page 17.)
- [Kajita 2001] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi and Hirohisa Hirukawa. *The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation*. In Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, volume 1, pages 239–246. IEEE, 2001. (Cited on page 8.)
- [Kalakrishnan 2010] Mrinal Kalakrishnan, Jonas Buchli, Peter Pastor, Michael Mistry and Stefan Schaal. *Fast, robust quadruped locomotion over challenging terrain*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 2665–2670. IEEE, 2010. (Cited on page 7.)
- [Kamedula 2018] Malgorzata Kamedula, Navvab Kashiri and Nikos G Tsagarakis. *On the kinematics of wheeled motion control of a hybrid wheeled-legged centauro robot*. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2426–2433. IEEE, 2018. (Cited on pages 7 and 11.)
- [Kamedula 2020] Malgorzata Kamedula, Navvab Kashiri and Nikos G Tsagarakis. *Wheeled motion kinematics and control of a hybrid mobility CENTAURO robot*. Robotics and Autonomous Systems, page 103482, 2020. (Cited on pages 7 and 11.)
- [Kanoun 2011] Oussama Kanoun, Florent Lamiraux and Pierre-Brice Wieber. *Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task*. IEEE Transactions on Robotics, vol. 27, no. 4, pages 785–792, 2011. (Cited on pages 15 and 46.)
- [KAWABATA 2014] Kentaro KAWABATA, Takuki NISHI, Yuki TORII, Masato KITAURA, Kota KANEMATSU and Sang-Ho HYON. *1A1-I02 Development of Hydraulic Quadruped Walking Robot RL-A1(Force Controlled Hydraulic Robots (1))*. The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec), vol. 33, 2014. (Cited on page 3.)
- [Khan 2015] Hamza Khan, Satoshi Kitano, Y Gao, Darwin G Caldwell and Claudio Semini. *Development of a lightweight on-board hydraulic system for a quadruped robot*. In 14th Scandinavian International Conference on Fluid Power-SICFP, 2015. (Cited on pages 3 and 4.)
- [Khatib 1986] 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. (Cited on pages 10 and 17.)
- [Khatib 1987] Oussama Khatib. *A unified approach for motion and force control of robot manipulators: The operational space formulation*. IEEE Journal on Robotics and Automation, vol. 3, no. 1, pages 43–53, 1987. (Cited on pages 12, 13, 16, 21 and 31.)
- [Khatib 2004a] Oussama Khatib. *Advanced robotic manipulation*, 2004. (Cited on page 16.)

- [Khatib 2004b] Oussama Khatib, Luis Sentis, Jaeheung Park and James Warren. *Whole-body dynamic behavior and control of human-like robots*. International Journal of Humanoid Robotics, vol. 1, no. 01, pages 29–43, 2004. (Cited on pages 12 and 17.)
- [Kim 2013] Jin Tak Kim, Jung San Cho, Byung-Yun Park, Sangdeok Park and Youngsoo Lee. *Experimental investigation on the design of leg for a hydraulic actuated quadruped robot*. In Robotics (ISR), 2013 44th International Symposium on, pages 1–5. IEEE, 2013. (Cited on pages 3 and 78.)
- [Klamt 2017] Tobias Klamt and Sven Behnke. *Anytime hybrid driving-stepping locomotion planning*. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4444–4451. IEEE, 2017. (Cited on page 11.)
- [Koolen 2016] Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas De Boer, Tingfan Wu, Jesper Smith, Johannes Engelsberger and Jerry Pratt. *Design of a momentum-based control framework and application to the humanoid robot atlas*. International Journal of Humanoid Robotics, vol. 13, no. 01, page 1650007, 2016. (Cited on pages 14 and 15.)
- [Kuindersma 2014] Scott Kuindersma, Frank Permenter and Russ Tedrake. *An efficiently solvable quadratic program for stabilizing dynamic locomotion*. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 2589–2594. IEEE, 2014. (Cited on page 15.)
- [Kuindersma 2016] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion and Russ Tedrake. *Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot*. Autonomous Robots, vol. 40, no. 3, pages 429–455, 2016. (Cited on page 15.)
- [Kumar 2019] Deepak Kumar and Somnath Sarangi. *Dynamic modeling of a dielectric elastomeric spherical actuator: an energy-based approach*. Soft Materials, pages 1–10, 2019. (Cited on page 81.)
- [Lee 2012] Sung-Hee Lee and Ambarish Goswami. *A momentum-based balance controller for humanoid robots on non-level and non-stationary ground*. Autonomous Robots, vol. 33, no. 4, pages 399–414, 2012. (Cited on pages 13, 98 and 113.)
- [Liu 2011] Chengju Liu, Qijun Chen and Danwei Wang. *CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 41, no. 3, pages 867–880, 2011. (Cited on page 9.)
- [Liu 2015] Yiping Liu, Patrick M Wensing, David E Orin and Yuan F Zheng. *Dynamic walking in a humanoid robot based on a 3d actuated dual-slip model*. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 5710–5717. IEEE, 2015. (Cited on page 9.)
- [Lubecki 2013] Tomasz Marek Lubecki. *Novel Energy Efficient Assistive Device with Intuitive Human-machine Interface*. PhD thesis, 2013. (Cited on page 17.)
- [Luh 1980] John YS Luh, Michael W Walker and Richard PC Paul. *On-line computational scheme for mechanical manipulators*. J. DYN. SYS. MEAS. & CONTR., vol. 102, no. 2, pages 69–76, 1980. (Cited on pages 13, 29, 76 and 82.)
- [Maes 2008] Ludovic D Maes, Marc Herbin, Rémi Hackert, Vincent L Bels and Anick Abourachid. *Steady locomotion in dogs: temporal and associated spatial coordination patterns and the effect of speed*. Journal of Experimental Biology, vol. 211, no. 1, pages 138–149, 2008. (Cited on page 88.)
- [Mason 2014] Sean Mason, Ludovic Righetti and Stefan Schaal. *Full dynamics LQR control of a humanoid robot: An experimental study on balancing and squatting*. In Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on, pages 374–379. IEEE, 2014. (Cited on pages 14 and 15.)

- [Mason 2016] Sean Mason, Nicholas Rotella, Stefan Schaal and Ludovic Righetti. *Balancing and walking using full dynamics LQR control with contact constraints*. In Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, pages 63–68. IEEE, 2016. (Cited on page 15.)
- [Mastalli 2016] Carlos Mastalli, Ioannis Havoutis, Michele Focchi, Darwin G Caldwell and Claudio Semini. *Hierarchical planning of dynamic movements without scheduled contact sequences*. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, pages 4636–4641. IEEE, 2016. (Cited on pages 10, 14 and 15.)
- [Mastalli 2020] Carlos Mastalli, Rohan Budhiraja, Wolfgang Merkt, Guilhem Saurel, Bilal Hamoud, Maximilien Naveau, Righetti Ludovic Carpentier Justin, Sethu Vijayakumar and Nicolas Mansard. *Crocodyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control*. In IEEE International Conference on Robotics and Automation (ICRA), 2020. (Cited on page 10.)
- [Melon 2020] Olivier Melon, Mathieu Geisert, David Surovik, Ioannis Havoutis and Maurice Fallon. *Reliable trajectories for dynamic quadrupeds using analytical costs and learned initializations*. arXiv preprint arXiv:2002.06719, 2020. (Cited on page 10.)
- [Mistry 2009] Michael Mistry, Stefan Schaal and Katsu Yamane. *Inertial parameter estimation of floating base humanoid systems using partial force sensing*. In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pages 492–497. IEEE, 2009. (Cited on page 12.)
- [Mistry 2010] Michael Mistry, Jonas Buchli and Stefan Schaal. *Inverse dynamics control of floating base systems using orthogonal decomposition*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 3406–3412. IEEE, 2010. (Cited on pages 12, 34 and 41.)
- [Morimoto 2008] Jun Morimoto, Gen Endo, Jun Nakanishi and Gordon Cheng. *A biologically inspired biped locomotion strategy for humanoid robots: Modulation of sinusoidal patterns by a coupled oscillator model*. IEEE Transactions on Robotics, vol. 24, no. 1, pages 185–191, 2008. (Cited on page 9.)
- [Nagano 2015] Kenta Nagano and Yasutaka Fujimoto. *The stable wheeled locomotion in low speed region for a wheel-legged mobile robot*. In 2015 IEEE International Conference on Mechatronics (ICM), pages 404–409. IEEE, 2015. (Cited on page 11.)
- [Nava 2016] Gabriele Nava, Francesco Romano, Francesco Nori and Daniele Pucci. *Stability analysis and design of momentum-based controllers for humanoid robots*. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pages 680–687. IEEE, 2016. (Cited on page 13.)
- [Nava 2018] Gabriele Nava, Luca Fiorio, Silvio Traversaro and Daniele Pucci. *Position and attitude control of an underactuated flying humanoid robot*. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), pages 1–9. IEEE, 2018. (Cited on page 99.)
- [Orin 2008] David E Orin and Ambarish Goswami. *Centroidal momentum matrix of a humanoid robot: Structure and properties*. In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, pages 653–659. IEEE, 2008. (Cited on pages xiii, 13 and 14.)
- [Orin 2013] 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. (Cited on pages 13, 14 and 22.)
- [Oriolo 2002] Giuseppe Oriolo, Alessandro De Luca and Marilena Vendittelli. *WMR control via dynamic feedback linearization: design, implementation, and experimental validation*. IEEE Transactions on control systems technology, vol. 10, no. 6, pages 835–852, 2002. (Cited on page 10.)



- [Orsolino 2018] Romeo Orsolino, Michele Focchi, Carlos Mastalli, Hongkai Dai, Darwin G Caldwell and Claudio Semini. *Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots*. IEEE Robotics and Automation Letters, vol. 3, no. 4, pages 3363–3370, 2018. (Cited on page 11.)
- [Ott 2008] Christian Ott, Alin Albu-Schaffer, Andreas Kugi and Gerd Hirzinger. *On the passivity-based impedance control of flexible joint robots*. IEEE Transactions on Robotics, vol. 24, no. 2, pages 416–429, 2008. (Cited on page 17.)
- [Ott 2010] Christian Ott, Ranjan Mukherjee and Yoshihiko Nakamura. *Unified impedance and admittance control*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 554–561. IEEE, 2010. (Cited on page 16.)
- [Ott 2011] Christian Ott, Maximo A Roa and Gerd Hirzinger. *Posture and balance control for biped robots based on contact force optimization*. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 26–33. IEEE, 2011. (Cited on page 13.)
- [Ott 2012] Christian Ott, Oliver Eiberger, Johannes Engelsberger, Maximo A Roa and Alin Albu-Schäffer. *Hardware and control concept for an experimental bipedal robot with joint torque sensors*. Journal of the Robotics Society of Japan, vol. 30, no. 4, pages 378–382, 2012. (Cited on page 17.)
- [Papadopoulos 1996] EG Papadopoulos and Daniel A Rey. *A new measure of tipover stability margin for mobile manipulators*. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, volume 4, pages 3111–3116. IEEE, 1996. (Cited on page 10.)
- [Park 2006] Jaeheung Park and Oussama Khatib. *Contact consistent control framework for humanoid robots*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 1963–1969. IEEE, 2006. (Cited on pages 12, 27, 32, 41 and 99.)
- [Park 2015] Hae-Won Park, Patrick M Wensing, Sangbae Kim *et al.* *Online planning for autonomous running jumps over obstacles in high-speed quadrupeds*. 2015. (Cited on page 9.)
- [Park 2017] Hae-Won Park, Patrick M Wensing and Sangbae Kim. *High-speed bounding with the MIT Cheetah 2: Control design and experiments*. The International Journal of Robotics Research, vol. 36, no. 2, pages 167–192, 2017. (Cited on page 9.)
- [Part 1985] SI Part. *Impedance control: An approach to manipulation*. Journal of dynamic systems, measurement, and control, vol. 107, page 17, 1985. (Cited on page 16.)
- [Ponce 2014] Oscar Efrain Ramos Ponce. *Generation of the whole-body motion for humanoid robots with the complete dynamics*. PhD thesis, Universite Toulouse III Paul Sabatier, 2014. (Cited on pages 15 and 17.)
- [Popovic 2005] Marko B Popovic, Ambarish Goswami and Hugh Herr. *Ground reference points in legged locomotion: Definitions, biological trajectories and control implications*. The international journal of robotics research, vol. 24, no. 12, pages 1013–1032, 2005. (Cited on page 11.)
- [Posa 2014] Michael Posa, Cecilia Cantu and Russ Tedrake. *A direct method for trajectory optimization of rigid bodies through contact*. The International Journal of Robotics Research, vol. 33, no. 1, pages 69–81, 2014. (Cited on page 10.)
- [Pratt 2001] Jerry Pratt, Chee-Meng Chew, Ann Torres, Peter Dilworth and Gill Pratt. *Virtual model control: An intuitive approach for bipedal locomotion*. The International Journal of Robotics Research, vol. 20, no. 2, pages 129–143, 2001. (Cited on page 16.)
- [Pucci 2016] Daniele Pucci, Gabriele Nava and Francesco Nori. *Automatic gain tuning of a momentum based balancing controller for humanoid robots*. In Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on, pages 158–164. IEEE, 2016. (Cited on pages 13 and 27.)

- [Raibert 1981] Marc H Raibert and John J Craig. *Hybrid position/force control of manipulators*. Journal of Dynamic Systems, Measurement, and Control, vol. 102, no. 127, pages 126–133, 1981. (Cited on page 16.)
- [Raibert 1984] Marc H Raibert, H Benjamin Brown Jr and Michael Chepponis. *Experiments in balance with a 3D one-legged hopping machine*. The International Journal of Robotics Research, vol. 3, no. 2, pages 75–92, 1984. (Cited on page 8.)
- [Raibert 1986] Marc H Raibert. Legged robots that balance. MIT press, 1986. (Cited on page 8.)
- [Raibert 1989] Marc H Raibert, H Benjamin Brown Jr, Michael Chepponis, Jeff Koechling, Jessica K Hodgins, Diane Dustman, W Kevin Brennan, David S Barrett, Clay M Thompson, John Daniell Hebert *et al.* *Dynamically Stable Legged Locomotion (September 1985-September 1989)*. 1989. (Cited on page 8.)
- [Raibert 2008] Marc Raibert, Kevin Blankespoor, Gabriel Nelson and Rob Playter. *Bigdog, the rough-terrain quadruped robot*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 10822–10825, 2008. (Cited on pages 3, 4 and 78.)
- [Riener 2005] Robert Riener, Lars Lunenburger, Saso Jezernik, Martin Anderschitz, Gery Colombo and Volker Dietz. *Patient-cooperative strategies for robot-aided treadmill training: first experimental results*. IEEE transactions on neural systems and rehabilitation engineering, vol. 13, no. 3, pages 380–394, 2005. (Cited on page 17.)
- [Righetti 2008] Ludovic Righetti and Auke Jan Ijspeert. *Pattern generators with sensory feedback for the control of quadruped locomotion*. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pages 819–824. IEEE, 2008. (Cited on page 9.)
- [Righetti 2011a] Ludovic Righetti, Jonas Buchli, Michael Mistry and Stefan Schaal. *Control of legged robots with optimal distribution of contact forces*. In Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, pages 318–324. IEEE, 2011. (Cited on page 13.)
- [Righetti 2011b] Ludovic Righetti, Jonas Buchli, Michael Mistry and Stefan Schaal. *Inverse dynamics control of floating-base robots with external constraints: A unified view*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1085–1090. IEEE, 2011. (Cited on pages 12, 13 and 34.)
- [Righetti 2012] Ludovic Righetti and Stefan Schaal. *Quadratic programming for inverse dynamics with optimal distribution of contact forces*. In Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on, pages 538–543. IEEE, 2012. (Cited on page 15.)
- [Righetti 2013] Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan and Stefan Schaal. *Optimal distribution of contact forces with inverse-dynamics control*. The International Journal of Robotics Research, vol. 32, no. 3, pages 280–298, 2013. (Cited on pages 13 and 15.)
- [Rollins 1998] Eric Rollins, J Luntz, Alex Foessel, Benjamin Shamah and William Whittaker. *Nomad: a demonstration of the transforming chassis*. In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, volume 1, pages 611–617. IEEE, 1998. (Cited on page 6.)
- [Saab 2013] Layale Saab, Oscar E Ramos, François Keith, Nicolas Mansard, Philippe Soueres and Jean-Yves Fourquet. *Dynamic whole-body motion generation under rigid contacts and other unilateral constraints*. IEEE Transactions on Robotics, vol. 29, no. 2, pages 346–362, 2013. (Cited on page 15.)
- [Saccon 2017] Alessandro Saccon, Silvio Traversaro, Francesco Nori and Henk Nijmeijer. *On Centroidal Dynamics and Integrability of Average Angular Velocity*. IEEE Robotics and Automation Letters, vol. 2, no. 2, pages 943–950, 2017. (Cited on page 99.)



- [Salini 2011] Joseph Salini, Vincent Padois and Philippe Bidaud. *Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1283–1290. IEEE, 2011. (Cited on pages 15 and 41.)
- [Saranlı 2010] Uluc Saranlı, Ömür Arslan, M Mert Ankaralı and Ömer Morgül. *Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping*. Nonlinear Dynamics, vol. 62, no. 4, pages 729–742, 2010. (Cited on page 8.)
- [Schwind 2000] William J Schwind and Daniel E Koditschek. *Approximating the stance map of a 2-DOF monoped runner*. Journal of Nonlinear Science, vol. 10, no. 5, pages 533–568, 2000. (Cited on page 8.)
- [Sciavicco 2014] L Sciavicco and L Villani. *Robotics: modelling, planning and control*. 2009, 2014. (Cited on pages 11 and 13.)
- [Semini 2010] Claudio Semini. *HyQ-design and development of a hydraulically actuated quadruped robot*. Doctor of Philosophy (Ph. D.), University of Genoa, Italy, 2010. (Cited on pages 3, 4 and 115.)
- [Semini 2016] Claudio Semini, Victor Barasuol, Thiago Boaventura, Marco Frigerio and Jonas Buchli. *Is active impedance the key to a breakthrough for legged robots?* In Robotics Research, pages 3–19. Springer, 2016. (Cited on pages xiii and 16.)
- [Semini 2017] Claudio Semini, Victor Barasuol, Jake Goldsmith, Marco Frigerio, Michele Focchi, Yifu Gao and Darwin G Caldwell. *Design of the Hydraulically Actuated, Torque-Controlled Quadruped Robot HyQ2Max*. IEEE/ASME Transactions on Mechatronics, vol. 22, no. 2, pages 635–646, 2017. (Cited on pages xiii, 3, 4, 17, 21, 22 and 115.)
- [Sentis 2005] Luis Sentis and Oussama Khatib. *Synthesis of whole-body behaviors through hierarchical control of behavioral primitives*. International Journal of Humanoid Robotics, vol. 2, no. 04, pages 505–518, 2005. (Cited on pages 11, 12, 17, 33, 35, 40, 42, 44 and 47.)
- [Sentis 2007] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, 2007. (Cited on pages 12, 21, 31 and 101.)
- [Sentis 2010] Luis Sentis, Jaeheung Park and Oussama Khatib. *Compliant control of multicontact and center-of-mass behaviors in humanoid robots*. IEEE Transactions on robotics, vol. 26, no. 3, pages 483–501, 2010. (Cited on pages 12, 33 and 102.)
- [Seok 2013] Sangok Seok, Albert Wang, Meng Yee Chuah, David Otten, Jeffrey Lang and Sangbae Kim. *Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 3307–3312. IEEE, 2013. (Cited on page 17.)
- [Seyfarth 2002] Andre Seyfarth and Hartmut Geyer. *Natural control of spring-like running-optimized self-stabilization*. In Proceedings of the Fifth International Conference on Climbing and Walking Robots. Professional Engineering Publishing Limited, pages 81–85, 2002. (Cited on page 8.)
- [Siciliano 2016] Bruno Siciliano and Oussama Khatib. Springer handbook of robotics. Springer, 2016. (Cited on pages 27, 30, 59 and 84.)
- [Siegwart 2002] Roland Siegwart, Pierre Lamon, Thomas Estier, Michel Lauria and Ralph Piguët. *Innovative design for wheeled locomotion in rough terrain*. Robotics and Autonomous systems, vol. 40, no. 2, pages 151–162, 2002. (Cited on page 6.)
- [Sreenivasan 1994] SV Sreenivasan and BH Wilcox. *Stability and traction control of an actively actuated micro-rover*. Journal of Field Robotics, vol. 11, no. 6, pages 487–502, 1994. (Cited on page 6.)

- [Suzumura 2013] Akihiro Suzumura and Yasutaka Fujimoto. *Real-time motion generation and control systems for high wheel-legged robot mobility*. IEEE Transactions on Industrial Electronics, vol. 61, no. 7, pages 3648–3659, 2013. (Cited on page 11.)
- [Thomson 2012] Travis Thomson, Inna Sharf and Blake Beckman. *Kinematic control and posture optimization of a redundantly actuated quadruped robot*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1895–1900. IEEE, 2012. (Cited on pages 6, 7 and 11.)
- [Traversaro 2017] Silvio Traversaro, Daniele Pucci and Francesco Nori. *A unified view of the equations of motion used for control design of humanoid robots*. Submitted to Multibody System Dynamics-Springer, 2017. (Cited on page 14.)
- [Tsagarakis 2017] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, L. Natale, E. Mingo Hoffman, H. Dallali, N. Kashiri, J. Malzahn, J. Lee, P. Kryczka, D. Kanoulas, M. Garabini, M. Catalano, M. Ferrati, V. Varricchio, L. Pallottino, C. Pavan, A. Bicchi, A. Settini, A. Rocchi and A. Ajoudani. *WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments*. Journal of Field Robotics, vol. 34, no. 7, pages 1225–1259, 2017. (Cited on page 17.)
- [Ugurlu 2010] Barkan Ugurlu and Atsuo Kawamura. *Eulerian ZMP resolution based bipedal walking: Discussions on the intrinsic angular momentum rate change about center of mass*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 4218–4223. IEEE, 2010. (Cited on page 13.)
- [Ugurlu 2015] Barkan Ugurlu, Ioannis Havoutis, Claudio Semini, Kana Kayamori, Darwin G Caldwell and Tatsuo Narikiyo. *Pattern generation and compliant feedback control for quadrupedal dynamic trot-walking locomotion: experiments on RoboCat-1 and HyQ*. Autonomous Robots, vol. 38, no. 4, pages 415–437, 2015. (Cited on page 14.)
- [Van Dijk 2015] W Van Dijk. *Human-Exoskeleton Interaction*. 2015. (Cited on page 17.)
- [Veneman 2007] Jan F Veneman, Rik Kruidhof, Edsko EG Hekman, Ralf Ekkelenkamp, Edwin HF Van Asseldonk and Herman Van Der Kooij. *Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 15, no. 3, pages 379–386, 2007. (Cited on page 17.)
- [Volpe 1996] Richard Volpe, J Balaram, Timothy Ohm and Robert Ivlev. *Rocky 7: A next generation mars rover prototype*. Advanced Robotics, vol. 11, no. 4, pages 341–358, 1996. (Cited on page 6.)
- [Vukobratović 2004] Miomir Vukobratović and Branislav Borovac. *Zero-moment pointthirty five years of its life*. International journal of humanoid robotics, vol. 1, no. 01, pages 157–173, 2004. (Cited on page 11.)
- [Walker 1982] Michael W Walker and David E Orin. *Efficient dynamic computer simulation of robotic mechanisms*. Journal of Dynamic Systems, Measurement, and Control, vol. 104, no. 3, pages 205–211, 1982. (Cited on pages 13, 29, 38 and 76.)
- [Wensing 2012] Patrick Wensing, Roy Featherstone and David E Orin. *A reduced-order recursive algorithm for the computation of the operational-space inertia matrix*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 4911–4917. IEEE, 2012. (Cited on page 13.)
- [Wensing 2013a] Patrick M Wensing and David E Orin. *Generation of dynamic humanoid behaviors through task-space control with conic optimization*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 3103–3109. IEEE, 2013. (Cited on pages 8, 14 and 15.)

- [Wensing 2013b] Patrick M Wensing and David E Orin. *High-speed humanoid running through control with a 3D-SLIP model*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 5134–5140. IEEE, 2013. (Cited on pages 8 and 14.)
- [Wensing 2014a] Patrick M Wensing. Optimization and control of dynamic humanoid running and jumping. The Ohio State University, 2014. (Cited on page 15.)
- [Wensing 2014b] Patrick M Wensing and David E Orin. *3D-slip steering for high-speed humanoid turns*. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 4008–4013. IEEE, 2014. (Cited on pages 8 and 14.)
- [Wensing 2014c] Patrick M Wensing and David E Orin. *Development of high-span running long jumps for humanoids*. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 222–227. IEEE, 2014. (Cited on page 8.)
- [Wensing 2015] Patrick M Wensing, Luther R Palmer and David E Orin. *Efficient recursive dynamics algorithms for operational-space control with application to legged locomotion*. Autonomous Robots, vol. 38, no. 4, pages 363–381, 2015. (Cited on pages 12 and 13.)
- [Wensing 2016] Patrick M Wensing and David E Orin. *Improved computation of the humanoid centroidal dynamics and application for whole-body control*. International Journal of Humanoid Robotics, vol. 13, no. 01, page 1550039, 2016. (Cited on pages 14, 22 and 30.)
- [Wensing 2017a] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang and S. Kim. *Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots*. IEEE Transactions on Robotics, vol. 33, no. 3, pages 509–522, June 2017. (Cited on pages 3, 4 and 5.)
- [Wensing 2017b] Patrick M Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang and Sangbae Kim. *Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots*. IEEE Transactions on Robotics, vol. 33, no. 3, pages 509–522, 2017. (Cited on page 80.)
- [Wieber 2006] Pierre-Brice Wieber. *Trajectory free linear model predictive control for stable walking in the presence of strong perturbations*. In 2006 6th IEEE-RAS International Conference on Humanoid Robots, pages 137–142. IEEE, 2006. (Cited on page 10.)
- [Winkler 2014] Alexander Winkler, Ioannis Havoutis, Stephane Bazeille, Jesus Ortiz, Michele Focchi, Rüdiger Dillmann, Darwin Caldwell and Claudio Semini. *Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots*. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 6476–6482. IEEE, 2014. (Cited on page 16.)
- [Winkler 2018] Alexander W Winkler, C Dario Bellicoso, Marco Hutter and Jonas Buchli. *Gait and trajectory optimization for legged systems through phase-based end-effector parameterization*. IEEE Robotics and Automation Letters, vol. 3, no. 3, pages 1560–1567, 2018. (Cited on page 10.)
- [Winter 1995] David A Winter. *Human balance and posture control during standing and walking*. Gait & posture, vol. 3, no. 4, pages 193–214, 1995. (Cited on page 8.)
- [Wu 2009] QiDi Wu, ChengJu Liu, JiaQi Zhang and QiJun Chen. *Survey of locomotion control of legged robots inspired by biological concept*. Science in China Series F: Information Sciences, vol. 52, no. 10, pages 1715–1729, 2009. (Cited on page 9.)
- [Yamane 2012] Katsu Yamane. *Systematic derivation of simplified dynamics for humanoid robots*. In Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on, pages 28–35. IEEE, 2012. (Cited on page 15.)

- [Yleonen 2002] S Yleonen and AARNE Halme. *Further Development and Testing of the Hybrid Locomotion of WorkPartner Robot*. In Proc. 5th International Conference on Climbing and Walking Robots (CLAWAR2002), pages 293–298, 2002. (Cited on pages 6 and 7.)
- [Yu 2012] Haitao Yu, Mantian Li, Pengfei Wang and Hegao Cai. *Approximate perturbation stance map of the SLIP runner and application to locomotion control*. Journal of Bionic Engineering, vol. 9, no. 4, pages 411–422, 2012. (Cited on pages xiii, 8 and 9.)
- [Yu 2017] Haitao Yu, Haibo Gao, Zeyang Fan, Zongquan Deng and Lixian Zhang. *Dual-SLIP model based galloping gait control for quadruped robot: A Task-space Formulation*. In American Control Conference (ACC), 2017, pages 191–197. IEEE, 2017. (Cited on page 9.)
- [Zeng 2019] Xuanqi Zeng, Songyuan Zhang, Hongji Zhang, Xu Li, Haitao Zhou and Yili Fu. *Leg Trajectory Planning for Quadruped Robots with High-Speed Trot Gait*. Applied Sciences, vol. 9, no. 7, page 1508, 2019. (Cited on page 78.)