Computationally-Robust and Efficient Prioritized Whole-Body Controller with Contact Constraints

D. Kim, J. Lee, J. Ahn, O. Campbell, H. Hwang, and L. Sentis[1]

Abstract— In this paper, we devise methods for the multi-objective control of humanoid robots, a.k.a. prioritized whole-body controllers, that achieve efficiency and robustness in the algorithmic computations. We use a form of whole-body controllers that is very general via incorporating centroidal momentum dynamics, operational task priorities, contact reaction forces, and internal force constraints. First, we achieve efficiency by solving a quadratic program that only involves the floating base dynamics and the reaction forces. Second, we achieve computational robustness by relaxing task accelerations such that they comply with friction cone constraints. Finally, we incorporate methods for smooth contact transitions to enhance the control of dynamic locomotion behaviors. The proposed methods are demonstrated both in simulation and in real experiments using a passive-ankle bipedal robot.

I. INTRODUCTION

During the DARPA Robotics Challenge in 2013 and 2015, various humanoid robots participating in the competition attempted to execute a variety of tasks using a class of controllers called whole-body controllers (WBC), which are easily reconfigurable to accomplish multiple control objectives under environmental constraints [1]. All WBCs for humanoid control share common features: accounting for system dynamics, addressing reaction forces to control the floating base, and using a mapping between the operational spaces and the configuration space. WBCs have become a standard method to control a multi-limb system executing both manipulation and locomotion. There are various types of WBCs, but for convenience, we categorize them as follows: 1) quadratic programming (QP) based WBCs without task hierarchy [2]–[5], 2) null space projection based WBCs [6]–[8], 3) hierarchical quadratic programming (HQP) [9]–[11], and 4) integrated quadratic programming and null space projection methods [12], [13].

A popular way to implement WBCs is using QPs to find the reaction forces and torque commands satisfying equality and inequality constraints. These methods have been widely adopted for real-hardware experiments. Using these algorithms on a compliant humanoid robot, [3] demonstrated the ability to lift a heavy object, balance under external disturbances, and walk. With the integration of a foot stepping planner, [14] showed robust stepping over cinder blocks and accomplished fast exploration of the rough terrain. [5] showed stable locomotion on cinder blocks and rubble. By integrating the WBC method and advanced contact area estimation, [15] accomplished walking using only partial footholds. In recent times, WBCs have

focused on leveraging early capabilities like task hierarchies, and increasing computational robustness and efficiency. Our methods shown here incorporate the latest features of WBCs while achieving higher computational efficiency than state-of-the-art methods.

Among the various capabilities of WBCs, a task hierarchy is important because it provides a guarantee that higher priority tasks will not fail due to conflicts with lower priority tasks. Hierarchies are similar to a resource allocation problem in which some resources (i.e. robot appendages or parts) are more important than others while attempting to accomplish multiple goals. Null space projection based methods accomplish a strict task hierarchy by projecting the lower priority task into the null space of higher priority tasks. This method was discussed early on by [16] and was extended to whole-body dynamic control in [6]. One drawback of projection-based methods is the inability to incorporate inequality constraints. This limitation becomes a concern when robot motions are highly dynamic causing the robot to break contact. Null space based methods assume that contact constraints are equalities which is physically incorrect.

HQP-based WBCs have been devised to enforce both task hierarchies and inequality constraints [9], [11], [17], [18]. [19] proposed prioritized task-space control by iteratively solving optimizations. [9] proposed an enhanced HQP by introducing a new QP formulation and a new solver that accomplishes WBC via a single QP problem instead of having multiple QPs running iteratively. [11] proposed efficient ways to solve iterative QPs and showed experimental results with a torque controlled robot. HQP-based WBC is a generic formulation that can incorporate many types of constraints and tasks. However, the computational cost is high for real-time applications (although several experimental validations have achieved about 1 ms computational times in full humanoid robots).

After previous studies [20], [21], we have learned that the feedback control performance heavily depends on control loop times. Bandwidth differences are more noticeable when the motion is highly dynamic. For example, in the demonstration of point-foot biped stabilization [20] using WBCs, increasing the update rate from 1kHz to 1.5kHz provided noticeably larger posture and foot position control bandwidth. Moreover, fast WBCs are also desirable when considering other problems that require heavy computations, such as model-predictive control, trajectory optimizations, and machine learning for motion generation. In this paper, we integrate QP-based methods and null space projection

[1] Luis Sentis is a Faculty of Aerospace Engineering, University of Texas at Austin, TX, USA lsentis@austin.utexas.edu

WBC methods within a formulation that achieves higher computational efficiency than leading WBC methods.

The approach presented in [12] is similar to ours, but we decrease the computational cost by removing operational space computations and instead computing acceleration in the configuration space based on the desired task commands. Also, the QP formulation in [12] can actually become infeasible due to torque limits and constraints, and their proposed method does not guarantee a strict task hierarchy. In contrast, our algorithm relaxes commanded task accelerations to make sure that the solution is always feasible. Furthermore, the torque limits in [12] can be violated because the torque commands for posture control are separated from the optimization, and added to the optimization result afterward.

For computational efficiency, we simplify the formulation even further over the previous acceleration-based methods by removing the terms representing the time derivative of the prioritized task Jacobians. Besides our efforts to increase computational efficiency, our formulation also addresses internal constraints and contact force transitions. By adding internal constraints, the torque command complies with constraints that internally exist such as mechanically coupled joints or closed kinematic chains. Our contact force transition method smooths out the jerk induced by a contact constraint change.

In summary, our contributions are: 1) a new WBC formulation, dubbed whole-body dynamic control (WBDC), providing benefits in efficient computation and algorithmic robustness, 2) the addition of new functionalities (internal constraints, contact transitions), and 3) the validation of the proposed algorithm both in physics-based simulations and in biped robot experiments.

II. WHOLE-BODY DYNAMIC CONTROL FORMULATION

We present an algorithm for whole-body control, dubbed WBDC, that improve over the state-of-the-art on efficiency and computational robustness. Fig. 1 illustrates the overall process performed by WBDC. The details are explained below.

A. Internal Constraints Inclusion

We prescribe constraints into two categories: 1) internal and 2) contact constraints. Internal constraints are holonomic and do not influence the floating base dynamics. Contact constraints involve reaction forces from the environment, and thus, exert forces on the floating base. Examples of internal constraints include bilateral constraints in coupled joints (e.g., hip yaw joints in the NAO [22] and waist joints in the Dreamer robot [23]) and closed chains in delta robots [24].

The first step in WBDC is projecting the multi-body dynamics into the null space of the internal constraints to remove the term representing the internal forces incurred by the constraint.

$$A\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = U^\top \boldsymbol{\tau} + J_{\text{int}}^\top \mathbf{F}_{\text{int}} + J_c^\top \mathbf{F}_r, \quad (1)$$

Internal Constraint Inclusion

$$A\ddot{\mathbf{q}} + N_{\text{int}}^\top(\mathbf{b} + \mathbf{g}) + J_{\text{int}}^\top(J_{\text{int}} A^{-1} J_{\text{int}}^\top)^{-1} \dot{J}_{\text{int}} \dot{\mathbf{q}}$$
$$= (U N_{\text{int}})^\top \boldsymbol{\tau} + (J_c N_{\text{int}})^\top \mathbf{F}_r$$

Section II-A

Contact Reaction Force Computation

1) Contact Constraints and First Task

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{c|\text{int}} + \overline{J_1 N_{c|\text{int}}}(\ddot{\mathbf{x}}_1^d - \dot{J}_1 \dot{\mathbf{q}} - J_1 \ddot{\mathbf{q}}_{c|\text{int}})$$

2) Reaction Force Optimization

$$\min_{\mathbf{F}_r, \boldsymbol{\delta}} \quad \mathbf{F}_r^\top Q_1 \mathbf{F}_r + \boldsymbol{\delta}^\top Q_2 \boldsymbol{\delta}$$

s.t.

$$\ddot{\mathbf{q}}_1 = \ddot{\mathbf{q}}_{c|\text{int}} + \overline{J_1 N_{c|\text{int}}}((\ddot{\mathbf{x}}_1^d + \boldsymbol{\delta}) - \dot{J}_1 \dot{\mathbf{q}} - J_1 \ddot{\mathbf{q}}_{c|\text{int}})$$
(first task execution)

$$S_f(A\ddot{\mathbf{q}}_1 + N_{\text{int}}^\top(\mathbf{b} + \mathbf{g})) = S_f J_{c|\text{int}}^\top \mathbf{F}_r$$
(floating base dyn.)

$$U_{aug} \mathbf{F}_r \geq \mathbf{0}, \quad \text{(contact force constraints)}$$

Section II-B

Prioritized Task Execution

$$\ddot{\mathbf{q}}_k = \ddot{\mathbf{q}}_{k-1} + \overline{J_{k|prec(k)}}\left(\ddot{\boldsymbol{\varphi}}_k^d - J_k \ddot{\mathbf{q}}_{k-1}\right)$$

Section II-C

$$\boldsymbol{\tau}$$
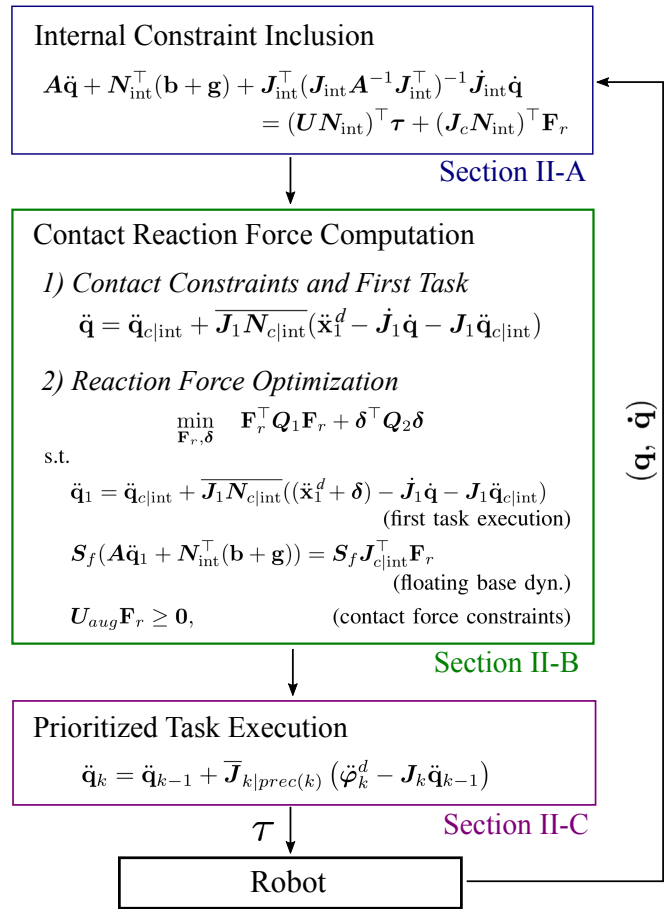
Robot

$$(\mathbf{q}, \dot{\mathbf{q}})$$

Fig. 1. **Whole-body Dynamic Control process flow.** During every control loop, the dynamic and kinematic model of a robotic system is updated with the sensed configuration $(\mathbf{q}, \dot{\mathbf{q}})$. WBDC projects dynamics into the null space of internal constraints (Section. II-A). After sequential projections addressing contact constraints and the first task are computed, a QP solver computes reaction forces and relaxed floating base task commands (Section. II-B). WBDC goes on to compute full configuration space accelerations by using a null space projection strategy. (Section. II-C).

where A, $\mathbf{b}$, and $\mathbf{g}$ represent mass, coriolis/centrifugal, and gravitational forces, respectively. U is a selection matrix to map actuated joint torques to the full configuration effort. J_{int} and J_c are internal and contact constraint Jacobians satisfying the conditions

$$0 = J_x \dot{\mathbf{q}},$$
$$0 = J_x \ddot{\mathbf{q}} + \dot{J}_x \dot{\mathbf{q}}, \quad (2)$$

where J_x represents either J_{int} or J_c. Before presenting the projected dynamics, we show two important terms. One is a dynamically consistent inverse,

$$\overline{J} = A^{-1} J^\top \left(J A^{-1} J^\top\right)^{-1}, \quad (3)$$

and the other one is a null space projection matrix,

$$N = I - \overline{J} J. \quad (4)$$

By using standard null space projection techniques, we

obtain the following equation,

$$A\ddot{\mathbf{q}} + N_{\text{int}}^\top(\mathbf{b} + \mathbf{g}) + J_{\text{int}}^\top(J_{\text{int}}A^{-1}J_{\text{int}}^\top)^{-1}\dot{J}_{\text{int}}\dot{\mathbf{q}}$$
$$= (UN_{\text{int}})^\top\tau + (J_cN_{\text{int}})^\top\mathbf{F}_r, \quad (5)$$

where N_{int} is the null space projection matrix derived from the internal constraints. Again, an important property of the internal constraints is that by definition they do not influence the floating base dynamics. Thus, the rows representing the floating base in $(UN_{\text{int}})^\top$ are all zero. Using this property, τ does not appear in the equality constraint of our QP formulation described in the next section.

B. Contact Reaction Forces Computation

Before finding reaction forces with QP, we need to address contact constraints. After computing the accelerations in the configuration space satisfying the contact constraints, the first task command is projected through the null space of the internal and contact constraints and added to the configuration space accelerations. Then, QP computes reaction forces satisfying unilateral and friction cone constraints. In the QP formulation, we add an upper bound on the reaction forces in the vertical direction to accomplish smooth contact force transitions.

1) Contact Constraints and First Task: Similar to the internal constraints, contact constraints are also characterized by zero acceleration at the contact points. Therefore, all task commands need to be in the null space of the contact constraints so that resultant motions do not incur acceleration at contact points. This is achieved by inverting the projected contact Jacobian, $\overline{J}_{c|\text{int}}(= \overline{J_cN_{\text{int}}})$, and from the condition in Eq. 2, we obtain

$$\ddot{\mathbf{q}}_{c|\text{int}} = \overline{J}_{c|\text{int}}(-\dot{J}_c\dot{\mathbf{q}}). \quad (6)$$

Then, we add the desired accelerations for the first task projected through the null space of the constraints,

$$\ddot{\mathbf{q}}_1 = \ddot{\mathbf{q}}_{c|\text{int}} + \overline{J_1N_{c|\text{int}}}(\ddot{\mathbf{x}}_1^d - \dot{J}_1\dot{\mathbf{q}} - J_1\ddot{\mathbf{q}}_{c|\text{int}}), \quad (7)$$

where $N_{c|\text{int}}(= I - \overline{J}_{c|\text{int}}J_{c|\text{int}})$ is a projection matrix of constraints and $\ddot{\mathbf{x}}_1^d$ is the acceleration command for the first task. $\dot{J}_1\dot{\mathbf{q}}$ and $J_1\ddot{\mathbf{q}}_{c|\text{int}}$ represent the influence on the first task acceleration caused by the configuration velocity and the computed acceleration incorporating constraints, respectively. Eq. (7) is similar to the equations in [13], but [13] omitted the term representing the influence of the higher priority task command to the lower priority, which is $J_1\ddot{\mathbf{q}}_{c|\text{int}}$ in Eq. (7).

Note that the first task must span centroidal dynamics, which is equivalent to spanning the floating base dynamics. Thus, in our simulations and experiments, we assign the first task to be either full joint position control, centroidal momentum control, or CoM and floating base orientation control task. To verify the first task spans the floating base dynamics, we check that the rank of the projected mass matrix is zero,

$$\text{rank}(S_fAN_{1|c|\text{int}}) = 0, \quad (8)$$

where S_f is a selection matrix to select the configuration representing the floating base. In general cases, the matrix selects the first six rows of A. $N_{1|c|\text{int}}$ is the null space projection matrix of the projected first task.

2) Reaction Force Computation: After obtaining the joint acceleration accounting for the constraints and the first task, we are now ready to compute the reaction forces. We use an open-source QP solver and follow the method proposed in [25]. To ensure that the QP problem is feasible, we relax the first task command. The formulation of our QP problem is

$$\min_{\mathbf{F}_r,\delta} \quad \mathbf{F}_r^\top Q_1\mathbf{F}_r + \delta^\top Q_2\delta \quad (9)$$

s.t.

$$\ddot{\mathbf{q}}_1 = \ddot{\mathbf{q}}_{c|\text{int}} + \overline{J_1N_{c|\text{int}}}\left((\ddot{\mathbf{x}}_1^d + \delta) - \dot{J}_1\dot{\mathbf{q}} - J_1\ddot{\mathbf{q}}_{c|\text{int}}\right)$$
(first task execution)
$$S_f\left(A\ddot{\mathbf{q}}_1 + N_{\text{int}}^\top(\mathbf{b} + \mathbf{g})\right) = S_fJ_{c|\text{int}}^\top\mathbf{F}_r$$
(floating base dyn.)
$$W_{aug}\mathbf{F}_r \geq \mathbf{0}, \quad \text{(contact force constraints)}$$

where $\mathbf{F}_r$, δ, and W_{aug} are reaction forces, relaxation variables, and an augmented contact constraint matrix, respectively. When describing the floating base dynamics, we remove the terms $J_{\text{int}}^\top(J_{\text{int}}A^{-1}J_{\text{int}}^\top)^{-1}\dot{J}_{\text{int}}\dot{\mathbf{q}}$ and $(UN_{\text{int}})^\top\tau$ of Eq. (5), since internal constraints have all zero terms in the rows corresponding to the floating base dynamics:

$$S_fJ_{\text{int}}^\top(J_{\text{int}}A^{-1}J_{\text{int}}^\top)^{-1}\dot{J}_{\text{int}} = \mathbf{0}$$
$$S_f(UN_{\text{int}})^\top = \mathbf{0}. \quad (10)$$

The contact force constraints in the QP shown in Eq. 9 include the friction cone inequalities to prevent slipping, and a reaction torque limit to prevent flipping or rolling the contacting link. In the case of a surface contact with a rectangular support area, the friction cone inequalities are

$$|f_x| \leq \mu f_z, \quad |f_y| \leq \mu f_z, \quad f_z > 0. \quad (11)$$

and the no-flip condition inequalities are

$$|\tau_x| \leq d_yf_z, \quad |\tau_y| \leq d_xf_z,$$
$$\tau_{z,\min} \leq \tau_z \leq \tau_{z,\max}, \quad (12)$$

where

$$\tau_{z,\min} \triangleq -\mu(d_x + d_y)f_z + |d_yf_x - \mu\tau_x| + |d_xf_y - \mu\tau_y|,$$
$$\tau_{z,\max} \triangleq \mu(d_x + d_y)f_z - |d_yf_x + \mu\tau_x| - |d_xf_y + \mu\tau_y|, \quad (13)$$

and d_x, d_y represent the distance from the center of the rectangle to the vertex in the local contact frame. Note that, $[\tau_x, \tau_y, \tau_z, f_x, f_y, f_z]^\top$ is the wrench representing resultant forces on the surface taken at the center of the rectangle in the local coordinate frame. [26] describes the closed-form formula for the contact wrench cone $W \in \mathbb{R}^{17\times 6}$. Then, the contact constraints for the i-th contact are compactly expressed as

$$W_i\mathbf{F}_r \geq \mathbf{0}. \quad (14)$$

The final step is to include contact frame rotations and augment to multiple contact points. For example, the formulation for k contact points is

$$\boldsymbol{W}_{aug}\boldsymbol{F}_r = \begin{bmatrix} \boldsymbol{W}_1\boldsymbol{R}_{\mathbf{aug,1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{W}_k\boldsymbol{R}_{\mathbf{aug,k}} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{r,1} \\ \vdots \\ \mathbf{F}_{r,k} \end{bmatrix} \geq \mathbf{0}, \tag{15}$$

where

$$\mathbf{R}_{\mathbf{aug,i}} = \begin{bmatrix} \boldsymbol{R}_i^{cm} & \mathbf{0}^{3\times3} \\ \mathbf{0}^{3\times3} & \boldsymbol{R}_i^{cm} \end{bmatrix}. \tag{16}$$

Note that $\boldsymbol{R}_i^{cm}$ is SO(3) from the CoM frame to the local contact coordinate frame and allows contact constraints to be expressed locally.

When selecting the weight matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ for the QP cost function in Eq. 9, we recommend weighting $\boldsymbol{Q}_2$ more heavily than $\boldsymbol{Q}_1$. For example, if we relax the height control task, then the QP will adjust the task to allow the robot to fall below the commanded height to reduce the vertical directional reaction forces because these forces are the largest numbers among the optimization variables in general.

3) Contact Constraint Transitions: WBDC also allows for optional smoothing of contact transitions by bounding the magnitude of reaction forces during the transition period. When using this functionality, we add the following inequality for the vertical directional reaction forces for each contact point:

$$\mathbf{F}_{r,z} \leq h\mathbf{F}_{r,z}^{\max} + (1-h)\mathbf{F}_{r,z}^{\min}, \tag{17}$$

where $h \in \begin{bmatrix} 0, & 1 \end{bmatrix}$. This constraint is included as an extra row in each $\boldsymbol{W}_i$ matrix shown in Eq 15. During the transition period, users increase or decrease h to avoid a sudden jump in the reaction forces when the contact is engaged or disengaged. Non-vertical reaction forces are bounded by the magnitude of the reaction force in the vertical direction, thus a single additional inequality constraint for each contact point is sufficient.

At this point the QP problem is fully specified. Running a solver will return the contact reaction forces $\mathbf{F}_r$ and the relaxation variables $\boldsymbol{\delta}$. Plugging the latter back into the *first task execution* constraint in Eq. 9 will yield $\ddot{\mathbf{q}}_1$.

C. Prioritized Task Executions

With $\ddot{\mathbf{q}}_1$, we can begin the process of iteratively projecting and applying the remaining prioritized tasks. Task-level controllers are computed in operational space as acceleration commands and converted to accelerations in configuration space using differential forward kinematics via

$$\dot{\mathbf{x}}_{\mathrm{task}} = \boldsymbol{J}_{\mathrm{task}}\dot{\mathbf{q}},$$
$$\ddot{\mathbf{x}}_{\mathrm{task}} = \boldsymbol{J}_{\mathrm{task}}\ddot{\mathbf{q}} + \dot{\boldsymbol{J}}_{\mathrm{task}}\dot{\mathbf{q}}, \tag{18}$$

where $\mathbf{x}_{\mathrm{task}} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^m$ represent the task's operational coordinates and the robot configuration, respectively, and $\boldsymbol{J}_{\mathrm{task}}$ is the corresponding Jacobian matrix. The joint acceleration resulting from the second desired task
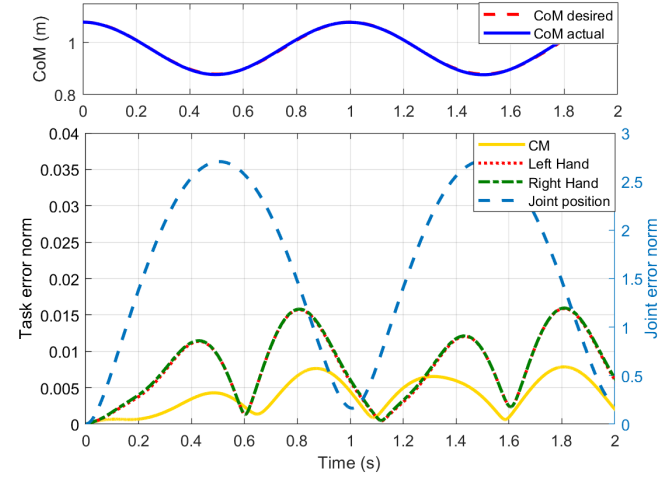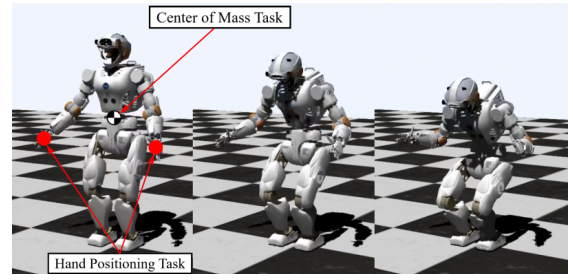

Fig. 2. **CoM height control while maintaining hand positions.** With both hand position fixed, Valkyrie moves up and down as the commanded CoM height changes. The data shows that Valkyrie's CoM height is almost identical to the desired trajectory. The norm of each task error becomes larger as the priority of the task becomes lower.

acceleration, $\ddot{\mathbf{x}}_2^d$ can be resolved as

$$\ddot{\mathbf{q}}_2 = \ddot{\mathbf{q}}_1 + \overline{\boldsymbol{J}_{2|1}}\left(\ddot{\boldsymbol{\varphi}}_2^d - \boldsymbol{J}_2\ddot{\mathbf{q}}_1\right), \tag{19}$$
$$\ddot{\boldsymbol{\varphi}}_2^d \triangleq \ddot{\mathbf{x}}_2^d - \dot{\boldsymbol{J}}_2\dot{\mathbf{q}}, \tag{20}$$

where $\overline{\boldsymbol{J}_{2|1}} \triangleq \overline{(\boldsymbol{J}_2\boldsymbol{N}_1)}$ represents the Jacobian associated with the second task, $\boldsymbol{J}_2$, projected into the null space of the first task, $\boldsymbol{N}_1 = \boldsymbol{I} - \overline{\boldsymbol{J}_1\boldsymbol{N}_{c|\mathrm{int}}}(\boldsymbol{J}_1\boldsymbol{N}_{c|\mathrm{int}})$. $\boldsymbol{N}_1$ by definition is orthogonal to the Jacobian associated with the projected first task, $\boldsymbol{J}_1\boldsymbol{N}_{c|\mathrm{int}}$. Eq. (19) can be extended to the general k-th task case, using the following hierarchy

$$\ddot{\mathbf{q}}_k = \ddot{\mathbf{q}}_{k-1} + \overline{\boldsymbol{J}}_{k|prec(k)}\left(\ddot{\boldsymbol{\varphi}}_k^d - \boldsymbol{J}_k\ddot{\mathbf{q}}_{k-1}\right) \tag{21}$$

with

$$\boldsymbol{J}_{k|prec(k)} = \boldsymbol{J}_k\boldsymbol{N}_{prec(k)},$$
$$\boldsymbol{N}_{prec(k)} = \prod_{s=2}^{k-1} \boldsymbol{N}_{s|prec(s)} \tag{22}$$
$$\boldsymbol{N}_{s|prec(s)} = \boldsymbol{I} - \overline{\boldsymbol{J}}_{s|prec(s)}\boldsymbol{J}_{s|prec(s)}$$

where $k \geq 2$, $\boldsymbol{N}_{prec(2)} = \boldsymbol{N}_1$. This task hierarchy is similar, but not identical to [13], [16]. Compared to those, our method is more concise, resulting in lower computation load for similar control specifications. In particular, we do not include the computation of time derivatives of prioritized Jacobians. Previous studies use $\dot{\boldsymbol{J}}_{k|prec(k)}\dot{\mathbf{q}}$ rather than $\dot{\boldsymbol{J}}_k\dot{\mathbf{q}}$,
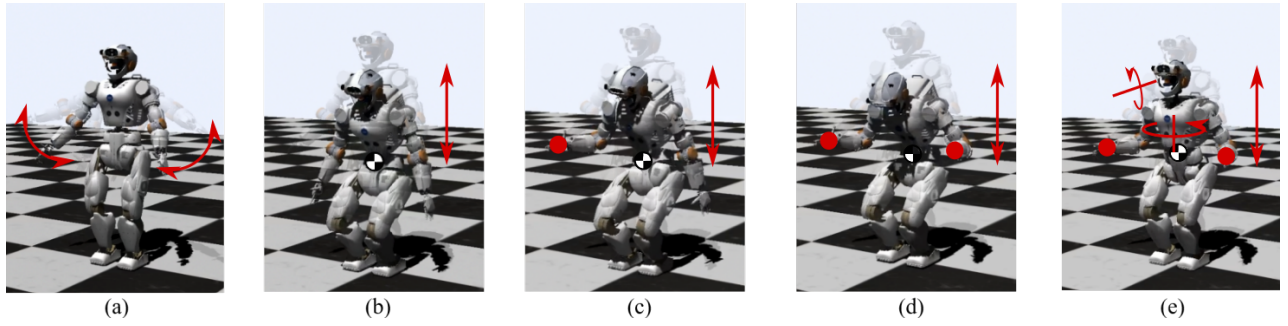
Fig. 3. **Implementation of various task sets:** Joint position (JP), Centroidal momentum (CM), Right hand position (RHP), Left hand position (LHP), Body orientation (BO), Head orientation (HO). (a) JP (b) CM + JP (c) CM + RHP + JP (d) CM + RHP + LHP + JP (e) CM + RHP + LHP + BO + HO + JP.
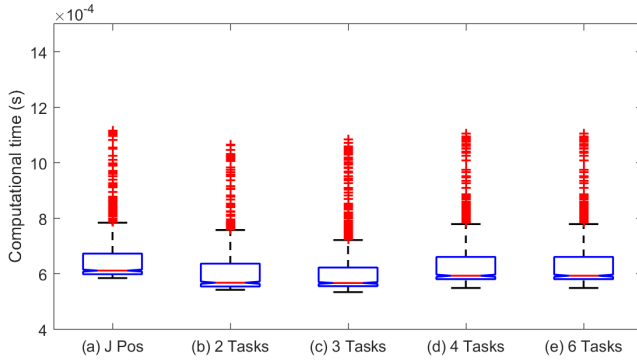

Fig. 4. **Computational time comparison:** The computation times of WBDC are plotted with respect to the task sets of the previous figure. 1000 iterations are considered for each case set.

which appears in Eq. (20). We use $\dot{J}_k \dot{q}$ because it correctly addresses the influence of the velocity on the task space.

The last step of WBDC is to compute a torque command from the contact reaction forces ($\mathbf{F}_r$) and the configuration space acceleration ($\ddot{q}$). By plugging these two terms into Eq. 5, we obtain

$$(\boldsymbol{U}\boldsymbol{N}_{\text{int}})^{\top}\boldsymbol{\tau} = \boldsymbol{A}\ddot{\mathbf{q}} + \boldsymbol{N}_{\text{int}}^{\top}(\mathbf{b} + \mathbf{g}) \\ + \boldsymbol{J}_{\text{int}}^{\top}(\boldsymbol{J}_{\text{int}}\boldsymbol{A}^{-1}\boldsymbol{J}_{\text{int}}^{\top})^{-1}\dot{\boldsymbol{J}}_{\text{int}}\dot{\mathbf{q}} - \boldsymbol{J}_c^{\top}\mathbf{F}_r. \quad (23)$$

Since all terms on the right hand side are known, we can calculate the torque command by multiplying the right hand side by the inverse of $(\boldsymbol{U}\boldsymbol{N}_{\text{int}})^{\top}$. To enhance the computational efficiency, we reduce the matrix by removing all zero terms corresponding to the floating base.

III. SIMULATION RESULTS

To verify the proposed methods, we perform physics-based simulations of Valkyrie, an adult size humanoid robot developed by NASA Johnson Space Center [27]. In our simulation, we reduced the number of DOFs of the robot by assuming that hand and wrist joints are fixed. Except for the reduced number of actuated joints, we built the Valkyrie simulation based on data provided by NASA. The simulations are designed to verify our claims: 1) execution of prioritized tasks, and 2) computational efficiency.

A. Multiple Task Execution

The proposed method is verified using complex whole body motions with double contact conditions as shown Fig. 2 and Fig. 3. In our simulation, Valkyrie has 28 actuated joints and the total dimensionality of the configuration space is 34 including the 6 virtual joints representing the floating base. The tasks used in the simulation are listed and the subscript indices indicate the priority of each task.

- $\ddot{\mathbf{x}}_1^d \in \mathbb{R}^6$: centroidal momentum (CM)
- $\ddot{\mathbf{x}}_2^d \in \mathbb{R}^3$: right hand position (RHP)
- $\ddot{\mathbf{x}}_3^d \in \mathbb{R}^3$: left hand position (LHP)
- $\ddot{\mathbf{x}}_4^d \in \mathbb{R}^{28}$: full joint posture (JP)

In Fig. 2, the CoM is commanded to move up and down while maintaining constant hand positions and double support contact. As explained in previous sections, the contact constraints are given the highest priority and the accelerations at the contact points are assumed to be zero. The first task is the CM task and is subordinate to the contact constraints. The CM task was designed to minimize the centroidal angular momentum while following a planned CoM trajectory. The vertical directional CoM trajectory is a sinusoidal function with 0.1m amplitude and 1.0Hz frequency. Next, the hand position tasks are defined to maintain a constant position. The last task specification is the joint position task for all joints and the corresponding desired values are the initial joint positions.

Fig. 2 shows the simulation results for the set of tasks defined above. As shown in the snapshots and the upper graph of Fig. 2, the robot successfully moves up and down to track the desired trajectory while maintaining constant hand positions. Since the sum of the task dimensions is larger than the total DOF of the robot, some lower priority tasks cannot be completely executed. For this reason, the joint position task with the lowest priority has significant error ($\|\mathbf{e}\|_2 \leq 3$), compared with errors of other tasks ($\|\mathbf{e}\|_2 \leq 0.02$) as shown in the second graph.

B. Computation Time Evaluation

To evaluate the computational performance of the proposed algorithm, we implemented five different simulation scenarios with varying numbers of tasks as shown in Fig. 3.

(a) Body Control

(b) Right Reaction Force

(c) Left Reaction Force

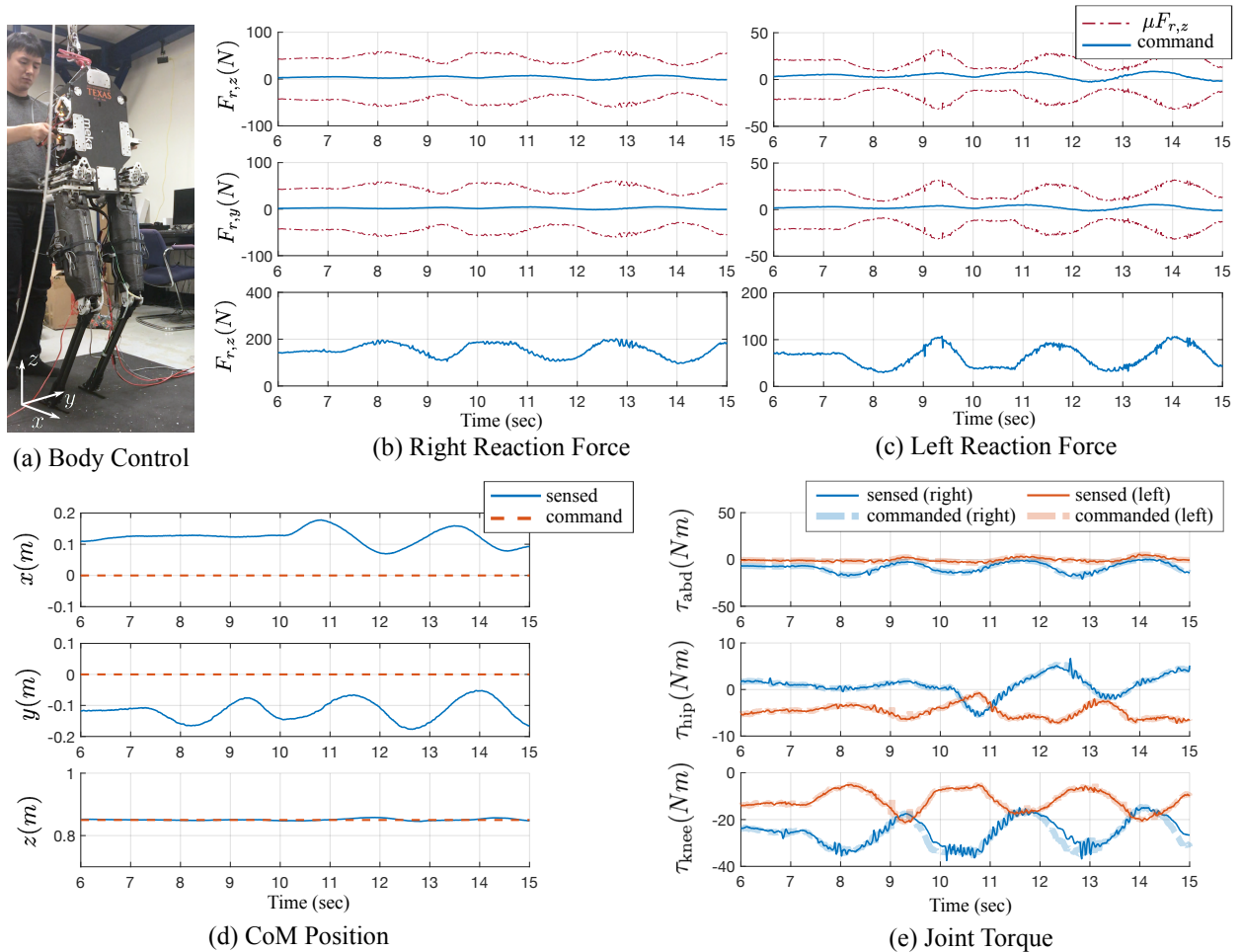(d) CoM Position

(e) Joint Torque

Fig. 5. **Body Pose Control Experiment.** (a) A person grabs handles of the body and move the body to the left and to the right. (b)(c) horizontal reaction forces satisfy friction cone constraints. The friction coefficient μ is set by 0.3. (d) x and y directional CoM position way off from the desired position control, which is an intended behavior. (e) DOB based torque feedback controller enables high accuracy torque tracking.

Tasks	mean / SD (ms)
JPos	0.632 / 0.0948
CM + JPos	0.612 / 0.0971
CM + RHP + JP	0.609 / 0.0999
CM + RHP + LHP + JP	0.636 / 0.0957
CM + RHP + LHP + BO + HO + JP	0.637 / 0.0957

TABLE I

COMPUTATION TIME EVALUATION

All of the scenarios use joint position as the lowest priority task, and each scenario successively incorporates one or two more tasks relative to the previous setup. We then timed our algorithm running on a dual-core 3.0 GHz Intel i7 processor for each of the five scenarios. Table I and Fig. 4 show the measured computation time for 1000 iterations of the algorithm for each simulated scenario.

Overall, the average computation time was around 0.6ms. In the proposed algorithm, the majority of the time is used to solve the QP, so the computation time would probably scale most noticeably with the size of the constraint set and the

cost function (which includes the first task). As shown in the Fig. 4, case (a) when the joint position task, which has the largest task dimension, is the first task incurs slightly more computational load than that of simulation case (b) when the CM task is the first task. Furthermore, since only the first task is included in the QP where most of the time is spent, the computation time does not increase significantly with respect to the number of hierarchical tasks from (b) to (e). Therefore, this algorithm works well even with a large number of prioritized tasks. In the current setup we did not incorporate the advanced decomposition algorithms used in [9], [11], so we believe that the computational efficiency could be improved by optimizing some of the matrix calculations.

IV. PASSIVE-ANKLE BIPED ROBOT EXPERIMENTS

Our robot, Mercury, is a biped robot that is 1.5 m tall and 23 kg in weight. The leg kinematics resemble simplified human kinematics and contain, in series, an adduction/abduction hip joint, a flexion/extension hip joint, and a flexion/extension knee joint, as shown in Fig. 5(a). The lack of ankle actuation allows the carbon fiber shank to be as lightweight as possible for quick foot movements. The
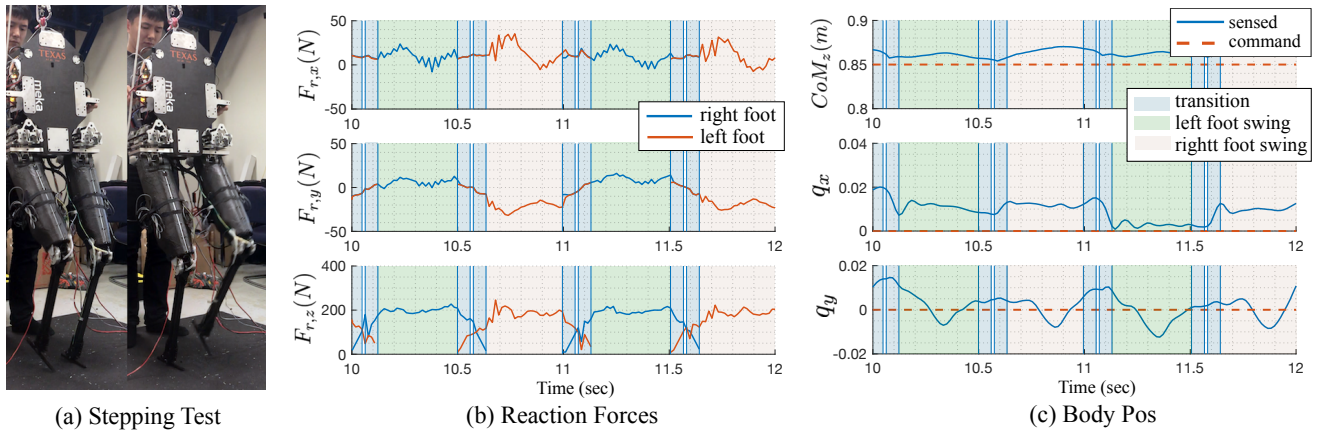
(a) Stepping Test   (b) Reaction Forces   (c) Body Pos

Fig. 6. **Stepping Test Experiment.** (a) During the test, a person gently holds the robot's body to constraint horizontal movements. (b) The vertical directional reaction forces smoothly change from zero to 200N during transition phases, which is the period colored by blue box. (c) Mercury's CoM height, roll, and pitch are maintained around the commanded pose. $\mathbf{q}_x$ and $\mathbf{q}_y$ mean x and y components of quaternion.

series elastic actuators (SEAs) on all six joints are based on a sliding spring carriage connected to the output joint by steel cables. The deformation in the springs is directly measured within the carriage assembly. For fall safety, the robot is attached to a pulley system with a block and tackle that allows us to lift the robot off of the ground. Mercury is controlled with distributed digital signal processors connected by an EtherCAT network to a centralized PC running a RT-Preempt Linux kernel. Each DSP controls a single actuator, and they do not communicate directly with each other. Power is delivered through a tether. The update frequencies for both WBDC and the torque feedback controllers are 1kHz, and should be faster in the near future.

In the experiments, we use our disturbance-observer (DOB) based torque feedback controller [28], which accomplishes both accurate torque tracking and large phase margin. High-fidelity joint torque control is important for successful implementations of WBC, especially for this robot since it can only obtain body posture control by exploiting dynamics (the lack of the ankle actuation does not allow the robot to stand statically).

A. Body Pose Control Test

In the first experiment, we used a single task to control body orientation and CoM position. The purpose of this test was to verify that the reaction forces satisfy the contact constraints and that WBDC controls the operational space task as we intended. As mentioned in Section. II-B.1, the first task needs to span the six floating base DOF (even though we know some of them are uncontrollable). We allocate small weights ($\boldsymbol{Q}_2$ in Eq. (9)) for CoM_x, CoM_y, and body yaw control to emphasize the importance of CoM height, body roll, and body pitch in the QP. During the experiment, a person lightly holds the robot torso and constrains the horizontal movements. Mercury controls the other aspects of its motion, such as height, roll, and pitch. To test the robot's compliance to horizontal disturbances, the experimenter moves the body left and right, and forward and backward (Fig. 5(a)).

The reaction forces presented in Fig. 5(b)(c) are commanded reaction forces since we cannot measure the reaction forces directly (the robot does not have force sensors at the feet). However, the commanded reaction forces satisfy the friction cone constraint as we can see in Fig. 5(b)(c). Fig. 5(e) presents the tracking performance of joint torque controller, and we can see the commanded and sensed torques almost overlap. The CoM position data in Fig. 5(d) shows that Mercury accurately maintains the commanded height. As we intended, the horizontal position is not restrained by the commanded position.

B. Stepping Test

The purpose of this test is to show how the contact transition method (Section. II-B.3) mitigates the jerk caused by a sudden change in the contact constraints while ensuring that our controller maintains the CoM height and body orientation during a stepping motion. Fig. 6(a) shows how we conducted the experiment. As we did in the body pose control experiment, a person gently holds the handles on the torso to constrain lateral movement. Stepping motion consists of three phases: double contact, transition, and swing. The duration of each phase was set to $0.055\mathrm{s}$ for transitions, $0.01\mathrm{s}$ for double contact, $0.36\mathrm{s}$ for swings.

Fig. 6(b) shows that the vertical reaction forces smoothly change from 0 to 200N during transition periods (blue box). Without the described transition smoothing, the reaction force commands will suddenly jump from 0 to 200N when contact constraints switch. The controlled position and orientation data are presented in Fig. 6(c). The x and y direction CoM and yaw orientation are not presented since they are uncontrolled dimensions. We use a quaternion for the body orientation description, so we show the q_x and q_y, which are x and y components of a quaternion $[q_\omega, q_x, q_y, q_z]$, instead of roll and pitch, respectively. We can see that the body orientation stays close to the commanded orientation during the stepping motion. The experimental results show that the proposed WBDC stably and accurately control stepping

motions, which is a promising foundation for the upcoming untethered walking tests.

V. Conclusion

In this paper, we propose a new WBC formulation that is computational efficient, algorithmically robust, and technically sound. The proposed algorithm ensures internal and contact constraints are satisfied while conforming to a strict task hierarchy. It uses QP to compute reaction forces satisfying contact constraints and then addresses operational space tasks defined as desired accelerations. Since we utilize acceleration-based prioritized task execution instead of computing operational space dynamics, the whole projection process is concise and efficient. In our physics-based simulation tests, WBDC computes torque commands for a 28-DoF humanoid robot in 0.6ms and the computation time is constant even when the number of tasks increases. The proposed methods were successfully implemented in our biped robot and the experiments demonstrated body posture control and smooth stepping behavior.

We are currently working on untethered walking with the passive ankle biped, which requires a robust planning algorithm and high-performance feedback controllers. By integrating the new WBDC and DOB-based torque feedback controller with our robust walking planning algorithm, we are aiming to show 3D untethered walking soon.

References

[1] K. Iagnemma and J. Overholt, *Special Issue: DARPA Robotics Challenge (DRC)*, J. Wiley and Sons, Eds. Journal of Field Robotics, 2015, vol. 32, no. 2.

[2] S. Kuindersma, F. Permenter, and R. Tedrake, "An efficiently solvable quadratic program for stabilizing dynamic locomotion," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2589–2594.

[3] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 1248–1255.

[4] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the darpa robotics challenge," *Journal of Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.

[5] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, pp. 1 650 007–35, Mar. 2016.

[6] L. Sentis, "Synthesis and control of whole-body behaviors in humanoid systems," Ph.D. dissertation, Stanford, 2007.

[7] L. Sentis, J. Park, and O. Khatib, "Compliant control of multicontact and center-of-mass behaviors in humanoid robots," *IEEE Transactions on robotics*, vol. 26, no. 3, pp. 483–501, 2010.

[8] Y. Lee, S. Hwang, and J. Park, "Balancing of humanoid robot using contact force/moment control by task-oriented whole body control framework," *Autonomous Robots*, vol. 40, no. 3, pp. 457–472, Oct. 2015.

[9] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[10] L. Righetti and S. Schaal, "Quadratic programming for inverse dynamics with optimal distribution of contact forces," in *12th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2012, pp. 538–543.

[11] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[12] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, p. 0278364916653815, Jul. 2016.

[13] L. Righetti and S. Schaal, "Quadratic programming for inverse dynamics with optimal distribution of contact forces," in *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2012, pp. 538–543.

[14] S. Kuindersma, R. Deits, M. Fallon, and A. Valenzuela, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2015.

[15] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt, "Walking on Partial Footholds Including Line Contacts with the Humanoid Robot Atlas," in *International Conference on Humanoid Robots*. Cancun: IEEE, Nov. 2016, pp. 1312–1319.

[16] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *5th International Conference on Advanced Robotics (ICAR), 'Robots in Unstructured Environments'*. IEEE, 1991, pp. 1211–1216.

[17] O. Kanoun, F. Lamiraux, and P. B. Wieber, "Kinematic Control of Redundant Manipulators: Generalizing the Task-Priority Framework to Inequality Task," *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 785–792, 2011.

[18] M. de Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 131, Jul. 2010.

[19] P. M. Wensing and D. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," *Robotics and Automation (ICRA)*, pp. 3103–3109, 2013.

[20] D. Kim, Y. Zhao, G. Thomas, B. R. Fernandez, and L. Sentis, "Stabilizing Series-Elastic Point-Foot Bipeds Using Whole-Body Operational Space Control," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1362–1379, 2016.

[21] Y. Zhao, N. Paine, and L. Sentis, "Feedback parameter selection for impedance control of series elastic actuators," *IEEE-RAS International Conference on Humanoid Robots*, pp. 999–1006, 2014.

[22] J. Strom, G. Slavov, and E. Chown, "Omnidirectional walking using zmp and preview control for the nao humanoid robot," in *Robot Soccer World Cup*. Springer, 2009, pp. 378–389.

[23] L. Sentis, J. Petersen, and R. Philippsen, "Implementation and stability analysis of prioritized whole-body compliant controllers on a wheeled humanoid robot in uneven terrains," *Autonomous Robots*, vol. 35, no. 4, pp. 301–319, 2013.

[24] F. Pierrot, C. Reynaud, and A. Fournier, "Delta: a simple and efficient parallel robot," *Robotica*, vol. 8, no. 2, pp. 105–109, 1990.

[25] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, Sep. 1983.

[26] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of Surface Contacts for Humanoid Robots: Closed-Form Formulae of the Contact Wrench Cone for Rectangular Support Areas," *arXiv.org*, Jan. 2015.

[27] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, and et al., "Valkyrie: NASA's First Bipedal Humanoid Robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, May 2015.

[28] D. Kim, J. Ahn, O. Campbell, N. Paine, and L. Sentis, "Investigations of a Robotic Testbed with Viscoelastic Liquid Cooled Actuators," *arXiv.org*, Nov. 2017.