



Versatile Locomotion Planning and Control for Humanoid Robots

Junhyeok Ahn^{1*}, Steven Jens Jorgensen², Seung Hyeon Bang³ and Luis Sentis³

¹Human Centered Robotics Lab, Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, United States, ²NASA Johnson Space Center, Houston, TX, United States, ³Human Centered Robotics Lab, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX, United States

We propose a locomotion framework for bipedal robots consisting of a new motion planning method, dubbed trajectory optimization for walking robots plus (TOWR+), and a new whole-body control method, dubbed implicit hierarchical whole-body controller (IHWBC). For versatility, we consider the use of a composite rigid body (CRB) model to optimize the robot's walking behavior. The proposed CRB model considers the floating base dynamics while accounting for the effects of the heavy distal mass of humanoids using a pre-trained centroidal inertia network. TOWR+ leverages the phase-based parameterization of its precursor, TOWR, and optimizes for base and end-effectors motions, feet contact wrenches, as well as contact timing and locations without the need to solve a complementary problem or integer program. The use of IHWBC enforces unilateral contact constraints (i.e., non-slip and non-penetration constraints) and a task hierarchy through the cost function, relaxing contact constraints and providing an implicit hierarchy between tasks. This controller provides additional flexibility and smooth task and contact transitions as applied to our 10 degree-of-freedom, line-feet biped robot DRACO. In addition, we introduce a new open-source and light-weight software architecture, dubbed planning and control (PnC), that implements and combines TOWR+ and IHWBC. PnC provides modularity, versatility, and scalability so that the provided modules can be interchanged with other motion planners and whole-body controllers and tested in an end-to-end manner. In the experimental section, we first analyze the performance of TOWR+ using various bipeds. We then demonstrate balancing behaviors on the DRACO hardware using the proposed IHWBC method. Finally, we integrate TOWR+ and IHWBC and demonstrate step-and-stop behaviors on the DRACO hardware.

OPEN ACCESS

Edited by:

Enrico Mingo Hoffman,
Italian Institute of Technology (IIT), Italy

Reviewed by:

Francesco Ruscelli,
Italian Institute of Technology (IIT), Italy

Yue Hu,

Tokyo University of Agriculture and
Technology, Japan

*Correspondence:

Junhyeok Ahn
junhyeokahn91@utexas.edu

Specialty section:

This article was submitted to
Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 20 May 2021

Accepted: 28 July 2021

Published: 13 August 2021

Citation:

Ahn J, Jorgensen SJ, Bang SH and
Sentis L (2021) Versatile Locomotion
Planning and Control for
Humanoid Robots.
Front. Robot. AI 8:712239.
doi: 10.3389/frobt.2021.712239

Keywords: humanoid robots, locomotion, trajectory optimization, whole body control, open source software

1 INTRODUCTION

Planning dynamically feasible motions for humanoid robots is a challenging problem. One difficulty arises from the need to use contact forces to control legged locomotion while conforming to terrain elevation and friction constraints. Robustly tracking planned motions in humanoids is yet another difficulty as they need to fulfill multiple task objectives and deal with redundancy and floating base dynamics. To facilitate humanoid mobility in diverse terrains and whole-body control for trajectory tracking, this paper proposes three methods: 1) a trajectory optimization for walking robots plus (TOWR+) for versatile dynamic locomotion planning, 2) an implicit hierarchical whole-body controller (IHWBC) for effective multi-objective trajectory tracking for legged systems, and 3) a

planning and control (PnC) software package to create complex mobility behaviors in diverse humanoid robots.

1.1 Dynamic Locomotion Planning for Legged Robots

We characterize different dynamic models used for locomotion planning purposes. Dynamic locomotion has been studied using point mass models (PM) with predefined footholds and step timings (Koolen et al., 2012; Engelsberger et al., 2015; Ahn et al., 2018) and PM models' variations to account for swing foot angular momentum (Faraji et al., 2019; Seyde et al., 2018; Ahn et al., 2020). These approaches are computationally efficient but require the contact schedule and the swing foot trajectories to be predefined by experts. This lack of automatic scheduling and foot swing trajectory optimization limits significantly their ability to realize complex behaviors.

To generate more complex and capable motions, Dai et al. (2014), Carpentier and Mansard (2018), Ponton et al. (2018) used a centroidal momentum (CM) model that projects all linkage motions onto a six-dimensional floating base, and has been used to optimize the momentum and reaction wrench trajectories with simplified feasibility assumptions about the robot kinematics and actuation. While Dai et al. (2014) incorporated linear complimentary formulations to automatically optimize contacts schedules, Kuindersma et al. (2016), Ponton et al. (2021) leveraged a separate contact planner formulated as an integer programming, to enable the optimization of contact sequences. This decoupling between the contact sequence optimization and motion optimization allows for reducing computational complexity but at a price: geometrically chosen contact sequences do not guarantee dynamical feasibility. In addition, the CM model does not include a robot's base orientation in its state vector, which is significantly limiting for motions that involve, for instance, turning. The CM model additionally requires a full-kinematics model for further optimization of the base orientation.

Hereid et al. (2016), Mastalli et al. (2020) used a full-order (FO) model, that includes both a floating base and joint dynamics, and considered to optimize the joint state and torque trajectories of robots. Mordatch et al. (2012), Posa et al. (2013) formulated multi-contact dynamics as a contact complementary problem and automatically optimize contact interactions with the environment while simultaneously considering joint trajectories. Although these approaches demonstrate complex locomotion behaviors and discovery of novel contact sequences, they are often incapable of being used for receding horizon control because of the complexity of the FO model.

As a moderate complexity option, Winkler et al. (2018) proposed a trajectory optimization for walking robots (TOWR) algorithm using a single rigid body (SRB) model. The SRB model assumes a constant inertia embedded in a CM model and allows for further optimization of the base orientation and feet position trajectories without involving a full-kinematic model. Moreover, TOWR optimizes contact sequences and timing without solving an integer programming or complementary problems. This framework is highly versatile to discover new gait patterns and motions with minimal predefined information yet is still computationally efficient thanks to a novel phase-based

TABLE 1 | Summary of properties of various legged robots. We have sampled 10^5 random configurations for all robots, computed the centroidal inertia tensor, and calculated the standard deviations for I_{xx} , I_{yy} , and I_{zz} .

	ANYmal B	Cassie	DRACO	ATLAS	Valkyrie
Foot type	Point	Line	Line	Surface	Surface
Leg DOFs	3	5	5	6	6
Leg weight (kg)	3.40	10.76	16.27	18.11	24.76
Inertia I_{xx}	0.15	0.26	1.36	3.81	4.69
Variation I_{yy}	0.43	0.23	0.81	3.25	4.14
(kg m ²) I_{zz}	0.44	0.22	1.37	2.86	2.41

parameterization. Unfortunately, the SRB model uses a constant inertia tensor for articulated robots without considering the limbs' inertia and assumes a robot morphology based on point-feet. While these are reasonable assumptions for quadrupeds, they oversimplify the dynamics of humanoids. **Table 1** summarizes some properties of various legged robots¹, where the quadruped ANYmal, has a point-feet with a light-weighted leg, while the compared humanoids have line-feet or surface-feet with significantly heavier legs. As such, the DRACO, ATLAS, and Valkyrie humanoids have larger centroidal inertia variations across its joint ranges and require a more delicate model than the SRB model².

To better handle dynamic motion planning of humanoids, this paper formulates a trajectory optimization algorithm, dubbed TOWR+, that leverages TOWR's phase-based parameterization but is tailored to humanoid and legged robots with wrench contact constraints and non-negligible centroidal inertia variations. To be more specific, we include reaction torque trajectories on the contact surfaces and end-effector orientation trajectories in the decision variables. This extension enables to consider humanoids with line or surface contacts and allows for the use of legs or other limbs with higher degrees-of-freedom than quadrupeds. In addition, we incorporate a composite rigid body (CRB) model to take the configuration dependent centroidal inertia into account. To compute the centroidal inertia without a full-kinematic model in the optimization process, we pre-train a regressor that maps floating base and end-effector configurations to the centroidal inertia tensor. We then provide an analytic expression of the trained network and its Jacobian to the optimizer.

1.2 Whole-Body Control

There are various types of WBC frameworks addressing the transcription of hierarchies between tasks, and unilateral contact constraints. Task hierarchies are important in WBCs to ensure that higher priority tasks will not fail due to conflict with lower priority tasks. There have been studies that impose a *strict hierarchy* on tasks, where Sentis and Khatib (2005), Kim D. et al. (2019), Kim et al. (2020) used null space projections and Saab et al. (2013), Herzog et al. (2016) solved a hierarchical quadratic program (QP). *Strict hierarchies* are attractive because

¹The properties of the robot are taken from the description files in the following repositories: https://github.com/ANYbotics/any_mal_b_simple_description (ANYmal B), https://github.com/erwincoumans/pybullet_robots (Cassie, ATLAS), <https://github.com/gkjohnson/nasa-urdf-robots> (Valkyrie), and <https://github.com/junhyeokahn/PnC> (DRACO).

²Exceptionally, Cassie has the leg actuators carefully located near the torso, which results in less variation of centroidal inertia being an atypical humanoid.

they allocate robots' resources (i.e., actuators) to accomplish higher priority tasks first and relax the enforcement of lower priority tasks. However, these techniques require an additional mechanism to compute a smooth and continuous transition between tasks as studied in Lee et al. (2012), Kim S et al. (2019). As another line of research, Wiedebach et al. (2016), Feng et al. (2014), Kuindersma et al. (2016), Apgar et al. (2018) imposed *implicit hierarchies* by directly describing the desired task space motions in the cost function with different weighting factors. Tasks weighted with larger weights result in higher priority than tasks with smaller weights. These WBCs lose the enforcement of strict dynamically consistent guarantees on the priorities, but win on being more intuitive and providing smoother transitions than WBCs with *strict hierarchies*.

WBCs typically enforce unilateral contact constraints to prevent contact penetration and slip, which can be expressed as zero accelerations of the contact bodies. Similarly, there are approaches that model contact constraints as a *hard constraint* through a null space projection technique (Sentis and Khatib, 2005; Saab et al., 2013; Kim et al., 2018) or equality constraint (Kuindersma et al., 2016; Koolen et al., 2016; Herzog et al., 2016). Although such approaches model exact unilateral constraints, they generate discontinuous torques during contact switches. As an alternative, Feng et al. (2014), Kim D. et al. (2019), Kim et al. (2020), Wiedebach et al. (2016), Apgar et al. (2018) modeled contact constraint as *soft constraint* by minimizing contact accelerations using weighting factors in the cost function. This allows for smooth contact switching through continuous interpolation of the weighting factors.

WBCs provide multi degrees-of-freedom control and are often integrated with a joint-level controller. For examples, Herzog et al. (2016), Kim et al. (2018) integrated their WBCs with a joint-level torque controller, while Feng et al. (2014), Kim et al. (2020) incorporated a joint-level position controller to handle model mismatch and provide stiff position behaviors. However, Feng et al. (2014), Kim et al. (2020) solved an inverse kinematics problem and compute joint kinematic targets, which can result in inconsistencies between the desired joint position commands and the dynamically consistent joint movements. To resolve this issue, Kuindersma et al. (2016), Koolen et al. (2016) instead integrated joint accelerations to obtain joint position commands.

In this paper we devise a new WBC framework, dubbed IHWBC. It employs an *implicit hierarchy* between multiple tasks and formulates unilateral contacts using *soft constraints* to provide continuous task and contact transitions, as well as more versatile task allocation. In addition to tracking the desired motion tasks, IHWBC tracks desired reaction wrench commands provided by motion planners and computes joint torque commands. It also computes joint position and velocity commands by integrating joint acceleration, which can then be fed to low-level joint position controllers. Although each individual feature embodied in IHWBC has been previously proposed in other WBC frameworks, this is the first time that we combine the best of all into a single IHWBC framework with an emphasis on versatility and ease of use for biped robots and humanoids.

In our previous work (Ahn et al., 2019; Kim et al., 2020), we successfully achieved unsupported dynamic stepping with DRACO using a PM model-based reactive footstep planner

and a projection-based WBC. However, the dynamic stability required continuous stepping without the ability to halt, which is not suitable for loco-manipulation tasks that require the robot to have zero velocity at the end of each movement (Jorgensen et al., 2020). Unlike other bipeds, DRACO has line-feet provided by five degrees-of-freedom per leg, and non-negligible mass distribution of its limbs. While similar to Cassie, DRACO has heavier legs and has up to 5x inertial variations as summarized in **Table 1**. For precise walking control of DRACO, we need to consider the leg's inertia and its angular momentum during fast swinging motions as reported by Kim et al. (2020), Wiedebach et al. (2016). In other words, we need to use an accurate model that can capture DRACO's heavy distal mass in motion planning as Faraji et al. (2019), Seyde et al. (2018) used a three-linear-pendulum and a lumped mass model, respectively. To further support our claims, **Figure 1** shows how a robot's mass distribution and foot size affect the performance of its walking and control. Taking into consideration all of this, we integrate the proposed TOWR+ planner and IHWBC controller and demonstrate it in the DRACO biped robot to balance and stop at every step.

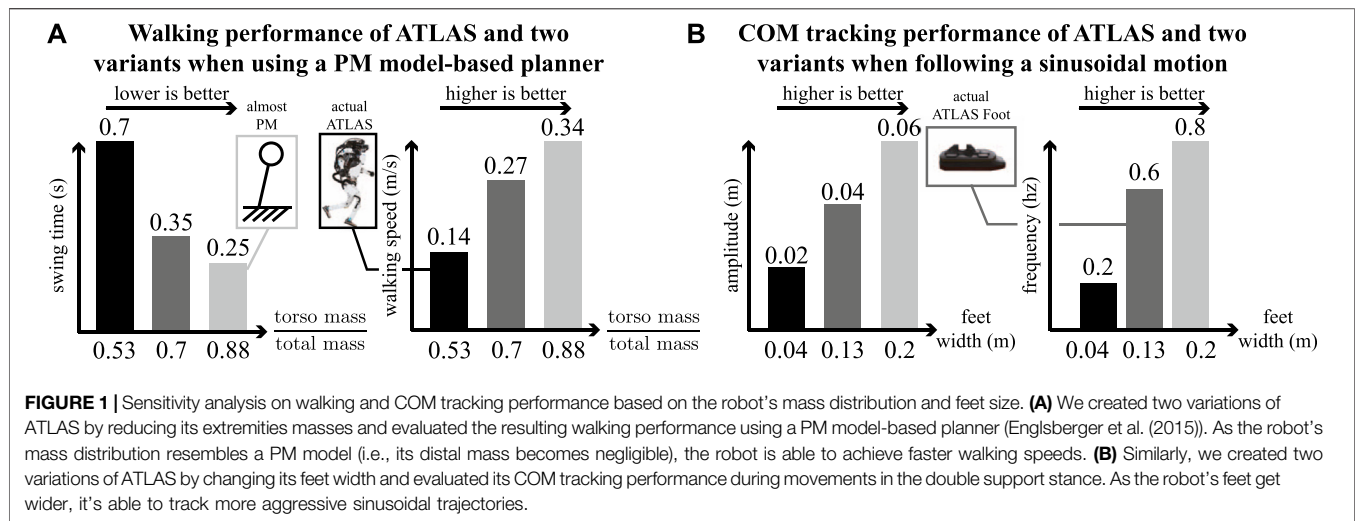
1.3 Open-Source Software Framework

Regarding locomotion planning, Crocoddyl (Mastalli et al., 2020), the multi-contact locomotion planning framework described in (Carpentier and Mansard, 2018), TOWR (Winkler et al., 2018), the kino-dynamic optimization framework described in (Ponton et al., 2021), and the inverted pendulum-based walking pattern generator described in (Caron et al., 2020) have been released open-source and maintained. Regarding WBC, the task space inverse dynamics framework described in (Del Prete et al., 2016), OpenSoT (Rocchi et al., 2015), mc_rtc (Bolotnikova et al., 2020), and mim_control (Grimminger et al., 2020) have been published open-source. The packages above focus either on locomotion planning or WBC but not on both. Therefore, these packages are not self-contained in that they require external libraries to fully plan and control humanoid robots. On the other hand, IHMC's open-robotics-software (IHMCRobotics, 2018) and MIT's Cheetah-software (Kim S et al., 2019) included both planning and WBC modules together and provide an end-to-end testing environment.

In this paper, we provide a software architecture, dubbed PnC, that implements our proposed TOWR+ motion planner and the IHWBC feedback controller. PnC implements the building blocks with sufficient abstraction of the tasks and constraints, such that it can be used with IHWBC but also with other WBC controllers or motion planners from third parties. Our software package contains various important modules such as physics simulators, a rigid body dynamics library, motion planners, and whole-body controllers so that users can connect their own motion planner or controller and evaluate it in an end-to-end manner with minimal external dependencies. Another unique feature of PnC is that we provide it in two different languages: a C++ version³ for those who are interested in the real-time performance, and a Python version⁴ for those who want to simply prototype and test their ideas.

³<https://github.com/junhyeokahn/PnC>

⁴<https://github.com/junhyeokahn/PyPnC>



1.4 Contribution

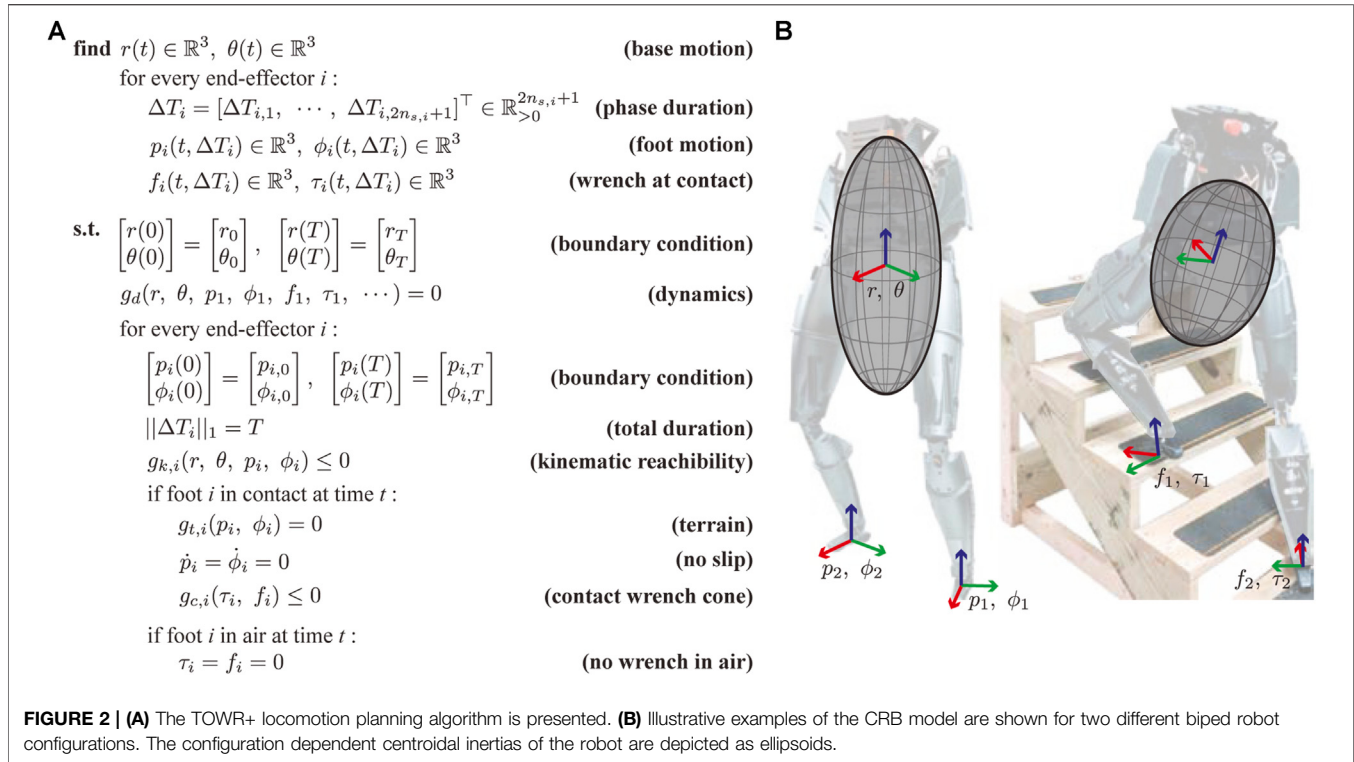
We highlight the key contributions of the paper:

- We propose a motion planning framework, dubbed TOWR+, that employs a CRB model and efficiently solves for robot body motions, end-effector motions and contact wrenches, and contact sequences, timings, positions and orientations. The CRB model is a simplified floating base model but is informative enough to consider the inertia effects of heavy robots limbs for planning thanks to the use of centroidal inertia network. We demonstrate various locomotion behaviors including a multi-contact case on multiple humanoid robots using TOWR+ with thorough analysis. We demonstrate that TOWR+ is able to discover arm motions to shape the robot's composite inertia while walking, which is a unique capability that other simplified model-based planners haven't achieved.
- We propose a whole-body controller, dubbed IHWBC, that considers an *implicit hierarchy* between tasks and *soft constraints* to account for unilateral contacts. IHWBC tracks the desired reaction wrench commands from our planner and computes joint torque commands. It also computes joint kinematic commands by integrating joint accelerations, which can be forwarded to low-level joint controllers. IHWBC allows smooth task and contact transitions, as well as a flexible task management, which is an important feature to control humanoid robots. We demonstrate experimental results on DRACO performing lateral swinging motion of its center of mass (COM) in double support phase by following a sinusoidal trajectory using IHWBC.
- We provide a lightweight and open-source software package, dubbed PnC, that integrates planning and control modules altogether, enabling an end-to-end evaluation environment with a focus on modularity, flexibility, and generality. The software package includes TOWR+ and IHWBC, but other algorithms can also be seamlessly integrated in the framework.

- Finally, we pair our TOWR+ and IHWBC components and evaluate locomotion behaviors under unknown disturbances in a physics simulator. We also experimentally demonstrate that DRACO can take a step and stop with zero velocity by capturing the full floating base state using our TOWR+ and IHWBC. To the best of our knowledge, this zero-velocity capture behavior on a line-foot biped with significant leg mass and centroidal inertia variations has not been achieved with simplified model-based planners and WBCs.

2 TOWR+

We propose a new motion planning framework dubbed TOWR+ for humanoids by leveraging the previous study, called TOWR, by Winkler et al. (2018). Our TOWR+ motion planning framework is shown in **Figure 2**. Similarly to TOWR, TOWR+ takes initial and final robot body states, the desired duration of the locomotion trajectory $T \in \mathbb{R}_{>0}$, the number of steps for each foot to reach the final robot state $n_{s,i} \in \mathbb{N}$, and the terrain elevation as an input. TOWR+ uses this information to find the COM trajectory $r(t) \in \mathbb{R}^3$, base orientation $\theta(t) \in \mathbb{R}^3$, and feet positions $p_i(t) \in \mathbb{R}^3$, orientations $\phi_i(t) \in \mathbb{R}^3$, reaction forces $f_i(t) \in \mathbb{R}^3$, and torques $\tau_i(t) \in \mathbb{R}^3$. Our algorithm automatically discovers appropriate gait schedules as defined by a phase duration vector $\Delta T_i \in \mathbb{R}_{>0}^{2n_{s,i}+1}$. Since each foot performs $n_{s,i}$ steps which includes a swing and a contact phase, the total number of contact phases for the foot is $2n_{s,i} + 1$. To ensure feasible motions, we employ a CRB model and simplified kinematic bounds for the feet positions. In addition, we impose additional constraints on the feet motion and contact wrenches to ensure that there is no slip on the feet and that wrenches are produced only while in contact. The decision variables and the constraints described in **Figure 2** are expressed in Cartesian space for ease of use. By extending TOWR we benefit by its computational advantages compared to other hybrid control methods that rely on complex integer programming or complementarity constraint specifications.



2.1 Phase-Based Parameterization

Similar to TOWR, we discretize the problem for numerical efficiency and we employ the above collocation method. We compose feet motions and wrench trajectories using multiple cubic Hermite polynomials, where each polynomial is defined by its duration and the positions and derivatives at the start and end of each node⁵. For instance, we discretize the base motion trajectories with a fixed timestep and parameterize their values and derivatives to solve the collocation problem. Along the base trajectories, we enforce continuous accelerations at the junction between polynomials, thus preventing sudden jumps on the contact points. As for foot motion trajectories, we use a predefined fixed number of polynomials during the swing phase and a constant position during the stance phase (since the feet are not moving). We also use a predefined fixed number of polynomials to compose contact wrench trajectories during the stance phase and use a zero wrench value during the swing phase. These parameterizations ensure that the feet only move during the swing phase and only generate reaction wrenches during the stance phase. We further constrain this processes to make sure that the feet motions and their wrench profiles are smooth and continuously differentiable during junctions. The duration of each phase is changed based on the optimized phase duration ΔT_i . Therefore, the duration of each polynomial during feet motions and contacts is automatically

determined through the optimization process, whereas the polynomial duration for base motions is fixed in advance. **Figure 3** shows an example of spline trajectories for some of the decision processes with their phase durations.

2.2 Robot Model

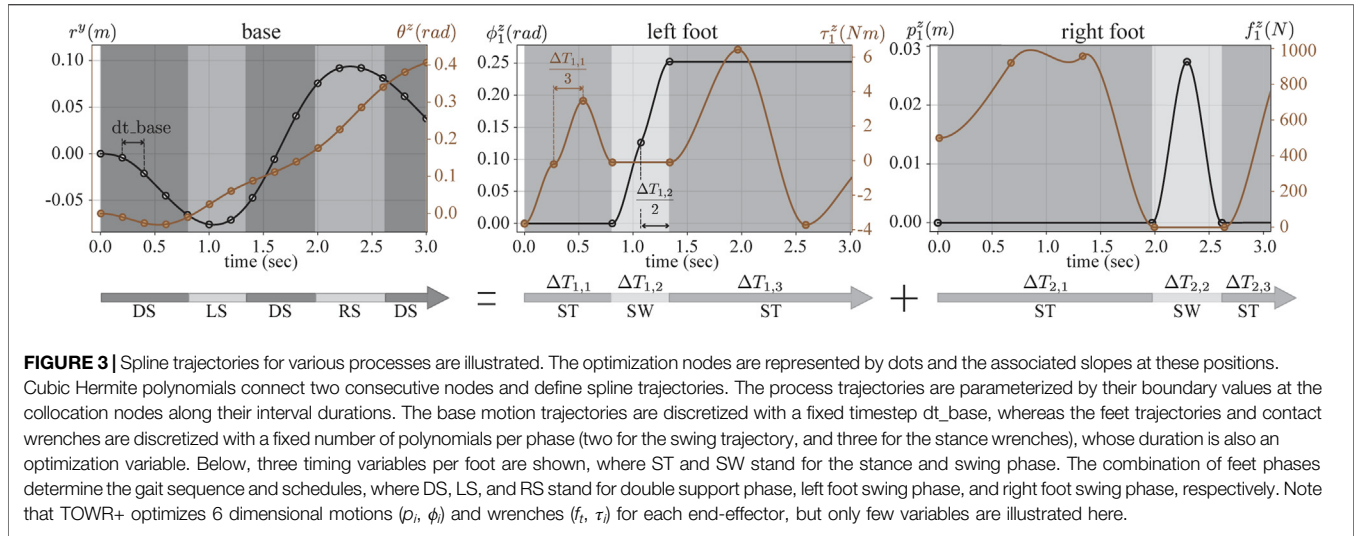
Here, we discuss a kinematic model for reachability and a new CRB model based on the use of a centroidal inertia network.

Kinematic model: We consider the relative distance between the robot's base and its feet to approximate the kinematic reachability constraint. Similarly to TOWR, we define the limits of the feet workspaces using cubes centered at nominal positions on each foot. As an extension in TOWR+, we further project each foot's heading (i.e., the foot frame's x -axis shown in **Figure 2**) onto the transverse plane (i.e., the base frame's x - y plane) and keep the angle between the projected foot heading and the base heading (i.e., base frame's x -axis) smaller than a threshold. Thus, we express the kinematic reachability constraint for the foot i , $g_{k,i}(r, \theta, p_i, \phi_i) \leq 0$ as:

$$\begin{aligned} & |R_b^w(\theta(t)) \underbrace{[p_i(t) - r(t)]}_{\dagger} - \tilde{p}_i| \leq l_{max}, \\ & \left| \angle \left(\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R_{ee_i}^b(\theta(t), \phi_i(t)) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\ddagger}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \right| \leq \theta_{max}, \end{aligned} \quad (1)$$

where R_b^w is the rotation matrix of the base frame from the world frame, $R_{ee_i}^b$ is the rotation matrix of the i th foot frame from the base frame, and $\tilde{p}_i$ represents the nominal position of the i th foot from

⁵Given pairs of foot positions, x_0 and x_1 , and their derivatives, $\dot{x}_0$ and $\dot{x}_1$, at the end points with time duration ΔT , a polynomial can be fully defined as $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$, where $a_0 = x_0$, $a_1 = \dot{x}_0$, $a_2 = -\Delta T^{-2}(3(x_0 - x_1) + \Delta T(2\dot{x}_0 + \dot{x}_1))$, and $a_3 = \Delta T^{-3}(2(x_0 - x_1) + \Delta T(\dot{x}_0 + \dot{x}_1))$.



the base frame. $\angle(\cdot, \cdot)$ measures the angle between two vectors. $l_{max} \in \mathbb{R}^3$ is the maximum deviation length vector from the nominal foot position, and $\theta_{max} \in \mathbb{R}$ is the maximum deviation angle of the projected foot heading from the base heading. Note that the expression, $\dagger$, represents the foot position relative to the base frame with respect to the base frame. The quantity, $\ddagger$, denotes the foot heading (i.e., x -axis) with respect to the base frame, which will then be projected onto the transverse plane. These kinematic constraints are enforced at sampled states while performing the trajectory optimization process.

Centroidal inertia network: The main motivation for incorporating a centroidal inertia along with the floating base model is to properly account for the heavy distal mass of most humanoid robots. However, considering a centroidal inertia with a floating base model is non-trivial because the centroidal inertia is a function of the robot's configuration (i.e., joint positions). To avoid solving an expensive inverse kinematic problem using the full kinematic model of a robot during the optimization process, we pre-train a centroidal inertia network that approximates the centroidal inertia tensor with respect to the base frame when the base and end-effectors configurations are given as an input with the expressions:

$$[I_{xx} \ I_{xy} \ I_{xz} \ I_{yy} \ I_{yz} \ I_{zz}]^T = I_b(r, \theta, p_1, \phi_1, \dots, p_{n_{ee}}, \phi_{n_{ee}}), \quad (2)$$

where I^* 's are the inertia quantities at the base frame and $I_b: \mathbb{R}^6 \times \mathbb{R}^{6 \times n_{ee}} \mapsto \mathbb{R}^6$ is the regressor function. The quantities r and θ represent the robot's base configuration and p_i and ϕ_i describe the i th end-effector configuration. One of the benefits of our composite centroidal inertia network is that it is possible to provide an analytic expression of the regressor I_b and its jacobian to the optimization solver, whereas inverse kinematic approaches do not provide analytic mappings.

For efficient training, we collect data from random sequences of stepping motions created using a few variables. For instance, we randomly sample initial and final configurations of the base, hands, and swinging foot, as well as stance foot configurations

and swing height positions. Then, we interpolate these initial and final configurations to generate a one-step motion using Cartesian frame samples. For all these interpolated trajectories, we collect n_{dpm} samples of the base and end-effectors configurations, solve full-body inverse kinematics to compute the robot's joint configurations along the trajectories, and compute centroidal inertia tensors. Repeating this procedure, we construct an input dataset using the base and end-effector configurations and label it using the exact centroidal inertia quantities. We then use this dataset to train the centroidal inertia network. The overall data generation pipeline and network structure are illustrated in **Figure 4**.

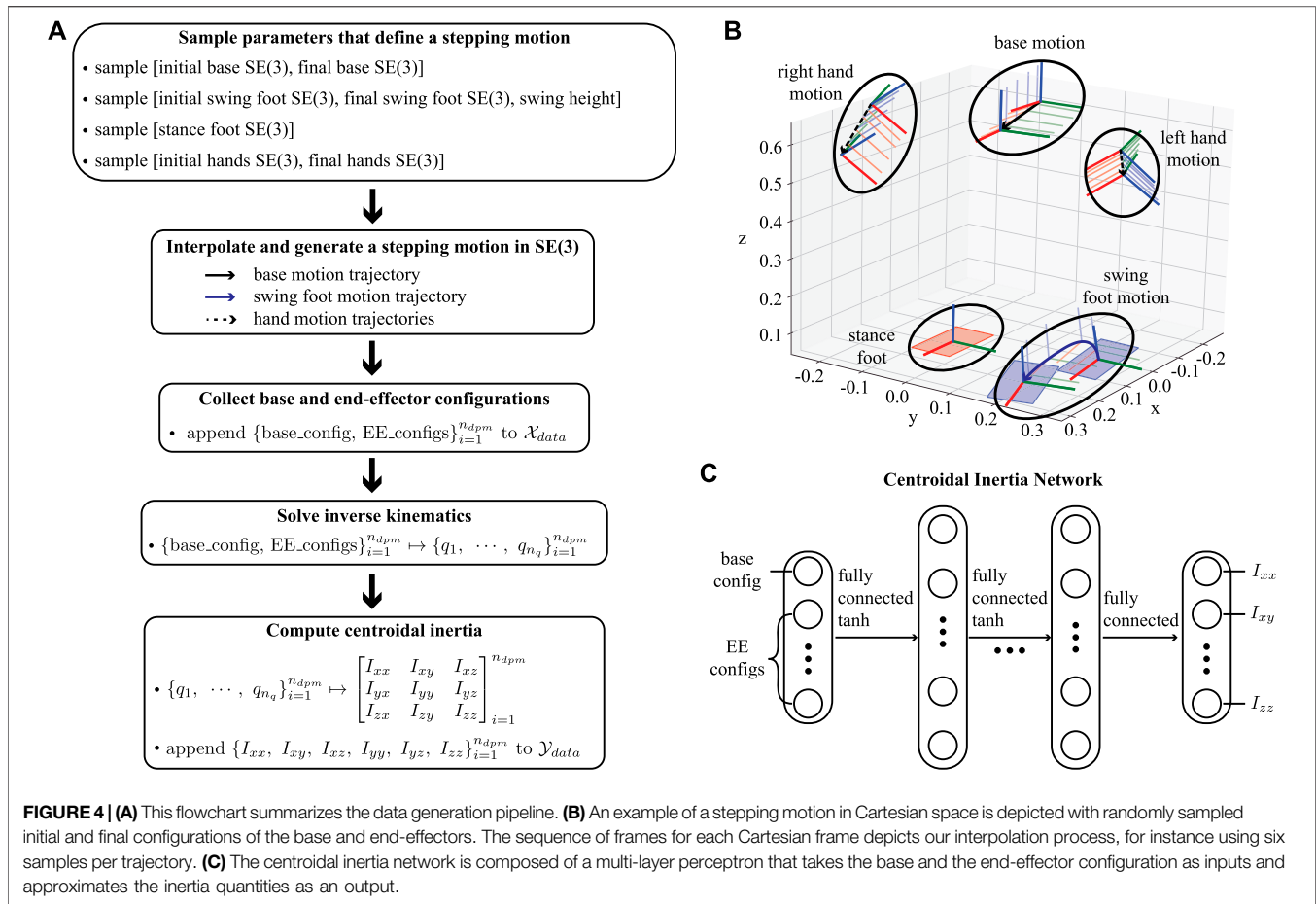
Composite rigid body model: Finally, we use the pre-trained centroidal inertia network above during the optimization process to enforce dynamic constraints. The base frame acceleration is defined by:

$$m\ddot{r}(t) = \sum_{i=1}^{n_{ee}} f_i(t) - mg, \\ I_w \dot{w}(t) + w(t) \times I_w w(t) = \sum_{i=1}^{n_{ee}} f_i(t) \times (r(t) - p_i(t)) + \tau_i, \quad (3)$$

where m is the mass of the robot, n_{ee} is the number of end-effectors, and g is the gravity vector. I_w is the centroidal inertia tensor expressed in world frame, which is efficiently predicted by the centroidal inertia network. Note that the centroidal inertia network, I_b , predicts the output with respect to the base frame, and we apply a transformation to express it in the world frame. $w(t)$ represents the angular velocity and can be calculated using the Euler angle rates $\dot{\theta}(t)$. We enforce the dynamic constraint at regular time intervals along the search.

2.3 Contact Model

As stated earlier, we use a phase-based parameterization to enforce no slip conditions and ensure zero contact wrenches while the foot is in the air. We optimize the duration of the polynomials, which determine the contact sequence and timings. For each foot's contact phase, we additionally consider the



contact wrench cone constraint and terrain elevation constraint. The contact wrench cone constraint, $g_{c,i}(\tau_i, f_i) = U_i[\tau_i(t), f_i(t)]^\top \leq 0$, where $U_i \in \mathbb{R}^{16 \times 6}$, ensures Coulomb friction fulfillment on the resultant force, the center of pressure of the foot being inside the support area, and the bounds on the yaw torque being fulfilled (Caron et al., 2015). The terrain elevation constraint enforces that the entire foot is in contact with the ground using the expressions:

$$\begin{aligned} [0 \ 0 \ 1] p_i(t) &= h_{\text{terrain}} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} p_i(t) \right), \\ R_{ee_i}^w(\phi_i(t)) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= n_{\text{terrain}} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} p_i(t) \right), \end{aligned} \quad (4)$$

where $R_{ee_i}^w$ represents the rotation matrix of the i th foot from the world frame. $h_{\text{terrain}}: \mathbb{R}^2 \mapsto \mathbb{R}$ is a function that receives a two dimension location of the terrain map as an input and returns its height (elevation), and $n_{\text{terrain}}: \mathbb{R}^2 \mapsto \mathbb{R}^3$ takes the same input and returns the normal vector of the terrain. Physically, these constraints ensure that the height of the foot matches the ground height and the contact surface is perpendicular to the terrain normal vector. Note that the contact wrench cone constraints and terrain elevation constraints are only enforced when the foot is in the contact phase and not enforced during the swing phase.

3 IMPLICIT HIERARCHICAL WHOLE-BODY CONTROL

We propose a new whole-body control framework, IHWBC, that employs an *implicit hierarchy* of tasks and *soft constraints* to handle unilateral contacts. This formulation enables smooth task and contact transitions, as well as flexible task management to be used in diverse legged and humanoid robots. Given desired task space objectives $\dot{x}_i^d$, and a stack of desired reaction forces f_r^d , the goal is to find instantaneous joint accelerations $\ddot{q}$ and dynamically consistent reaction wrenches f_r , that satisfy the robot's kinematics and dynamics constraints. The optimal solutions $\ddot{q}_*$ and f_r^* are then used to compute desired joint torques τ using forward dynamics. The joint accelerations $\ddot{q}_*$ are subsequently integrated to obtain desired joint velocities and positions that are then sent to a low-level joint controller in the robot. As noted by Feng et al. (2014), care must be taken when integrating joint accelerations to prevent instability problems, such as wind-up. We follow the integration approaches described in IHMC Robotics (2018), Jorgensen (2020). Our IHWBC is formulated as a linear constrained QP:

$$\min_{\ddot{q}, f_r} \sum_{i=1}^n w_i \|\ddot{q}_i - \ddot{x}_i^d\|^2 + w_f \|f_r^d - f_r\|^2 + \lambda_q \|\ddot{q}\|^2 + \lambda_f \|f_r\|^2 \quad (5A)$$

$$\text{s.t.} \quad S_f(A\ddot{q} + b + g - J_c^\top f_r) = 0, \quad (5B)$$

$$Uf_r \geq 0, \quad (5C)$$

$$S_r f_r \leq f_r^{\max}, \quad (5D)$$

$$\ddot{q}_{\min} \leq \ddot{q} \leq \ddot{q}_{\max}, \quad (5E)$$

$$\tau_{\min} \leq S_a (A\ddot{q} + b + g - J_c^T f_r) \leq \tau_{\max}. \quad (5F)$$

We now discuss the individual terms of the above optimization.

Task space acceleration objectives: The first term in **Eq. 5A** describes task space acceleration objectives. The i th task space acceleration, $\ddot{x}_i^d$, can be computed using simple task space PD feedback, i.e., $\ddot{x}_i^d = k_{p,i}(x_i^d - x_i) + k_{d,i}(\dot{x}_i^d - \dot{x}_i)$, where, $k_{p,i}$ and $k_{d,i}$ are PD gains. The desired task space position and velocity, x_i^d and $\dot{x}_i^d$, are provided by TOWR+ or other motion planners, and the measured position and velocity, i.e., x_i and $\dot{x}_i$, is obtained via forward kinematics from joint and IMU sensor data. $\dot{q}_m$ corresponds to measured joint velocities. Each task has a priority weight, w_i , and uses a task Jacobian J_i to establish a correspondence with generalized displacements. Unilateral contact constraints are accounted for by penalizing contact accelerations through assigning high values to their weights. Our strategy to incorporate non-slip constraints is to set the desired positions and velocities as the current ones (i.e., $x_i^d = x_i$ and $\dot{x}_i^d = \dot{x}_i$). Placing a high weight value on contact tasks is akin to projecting lower priority tasks in the contact null space. In summary, our new IHWBC imposes an *implicit hierarchy* between tasks through the use of task weights and provides *soft constraints* to handle contacts.

Smooth task and contact transitions: Using IHWBC, we guarantee smooth task transitions by changing the relative weights w_i for all tasks i . To transition from an old task weight, w_i^{old} , to a target task weight, w_i^{target} , we perform the following linear interpolation:

$$w_i^{\text{update}} = (1 - s(t))w_i^{\text{old}} + s(t)w_i^{\text{target}} \quad (6)$$

where $s: \mathbb{R} \rightarrow [0, 1]$ is a monotonic scalar function of time, t , whose output goes from 0 to 1. For instance, given a task transition time, $T_{\text{transition}}$, $s(t)$ is defined as:

$$s(t) = \frac{t}{T_{\text{transition}}}. \quad (7)$$

Desired reaction wrenches: The next term in the cost function, **Eq. 5A**, tracks desired reaction forces, f_r^d . A scalar weight w_f is provided to indicate the relative priority of the force tracking task compared to tracking task space acceleration tasks. This reaction force tracking capability is used to track forces provided by motion planners, such as feet reaction forces or desired manipulation forces.

Regularization and smoothing terms: Regularization costs λ_q and λ_f in **Eq. 5A** are incorporated to the decision variables, q and f_r , respectively, to ensure that the QP cost terms are positive definite as certain configurations, q , can cause task Jacobians, J_i , to be near singularities. Without these regularization terms, the QP can encounter numerical issues as some QP solvers require the quadratic cost matrix to be strictly convex (Goldfarb and Idnani, 1983; Turlach and Weingessel, 2011).

Dynamics constraint on floating base: The equality constraint **Eq. 5B** enforces floating base dynamics, where S_f is a selection matrix that extracts the first six rows of the generalized dynamics equation. A , b ,

g , and J_c are the mass matrix, centrifugal and coriolis force, gravity vector, and the concatenation of contact Jacobians, respectively.

Contact wrench cone and maximum force constraints: The constraints defined by **Eqs 5C,D** are the contact wrench cone and maximum force constraints, respectively. The matrix U changes depending on the contact type. Typically, the types of contacts include point, line, or surface contacts (Park and Khatib, 2008). The matrix S_r is a selection matrix used to bound a specific reaction force to a maximum value described by the maximum force vector $f_r^{\max}$. In practice, we select the normal direction of the reaction forces and bound them from above while bounding the tangential force directions using the cone constraints. When a robot makes or breaks contact, we smoothly increase or decrease the upper bound so that the robot has smooth transitions. Note that for each non-slip contact constraint that is defined, we need to incorporate a position task defined at the contact frame to ensure near zero accelerations.

Kinematic constraints: We also specify minimum and maximum joint acceleration limits as shown in **Eq. 5E**. Alternatively, joint acceleration bounds can be enforced using position and velocity limits. For instance, assuming an integration time of Δt and using Euler integration, the acceleration limits are bounded by the joint position and velocity limits as:

$$\dot{q}_{\min} \leq \dot{q}\Delta t + \dot{q}_m \leq \dot{q}_{\max}, \quad q_{\min} \leq \frac{1}{2}\ddot{q}\Delta t^2 + \dot{q}_m\Delta t + q_m \leq q_{\max}, \quad (8)$$

where $\dot{q}_m$ and q_m are the measured joint velocity and position, respectively. A conservative limit can also be enforced by over-estimating the integration time Δt . To account for all possible joint and velocity limits expressed as joint acceleration constraints, the reader can refer to Del Prete (2018).

Torque limits: Actuator saturation constraints are defined in **Eq. 5F**, where S_a is a selection matrix that extracts the bottom rows of the generalized dynamics equation corresponding to actuated joints. Once the QP solution $\ddot{q}^*$ and f_r^* are obtained, the optimal joint torque command can be obtained using:

$$\tau^{\text{cmd}} = S_a (A\ddot{q}^* + b + g - J_c^T f_r^*). \quad (9)$$

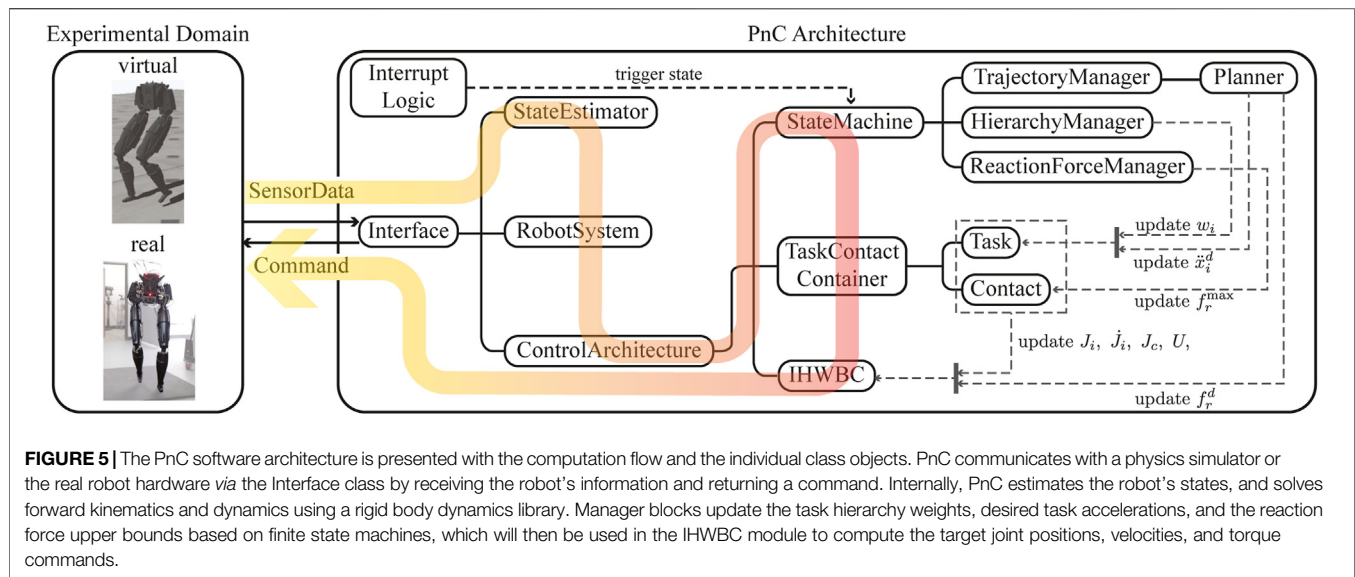
Using the integration schemes described in IHMC Robotics (2018), Jorgensen (2020) we then compute q^{cmd} and $\dot{q}^{\text{cmd}}$, which will then be executed by a low-level joint position controller. The torque input for each joint can be computed as:

$$\tau^{\text{joint}} = \tau^{\text{cmd}} + k_p(q^{\text{cmd}} - q_m) + k_d(\dot{q}^{\text{cmd}} - \dot{q}_m), \quad (10)$$

where k_p and k_d are the PD gains of the low-level controller.

4 PNC SOFTWARE ARCHITECTURE

We introduce a new open-source and light-weight software architecture, dubbed PnC, that implements and combines TOWR+ and IHWBC with control tools to design new behaviors. The main goal of PnC is to provide various modules for planning, whole-body control, simulation, rigid body dynamics, and utilities, such that they can be executed with minimal external dependencies. **Figure 5** shows the overall PnC architecture described in more detail below.



Interface: PnC can be seamlessly employed for real-time experiments or virtual simulations. The communication is handled by the Interface class which takes *SensorData* inputs and returns *Command* as computed by PnC. Here, *SensorData* contains joint encoder measurements, force and torque data, IMU data, and camera data, and *Command* consists of desired joint positions, velocities, and torques to be applied to the robot. This computational cycle takes place at every servo loop for simulation or real robot experimentation. Currently, PnC uses the PyBullet simulator (Coumans and Bai, 2016–2019).

RobotSystem: The *StateEstimator* class computes the floating base state estimation using *SensorData* and updates the internal robot model using *RobotSystem* class. *RobotSystem* is a wrapper class for rigid body dynamics, such as interfacing with Dart (Lee et al., 2018) or Pinocchio (Carpentier et al., 2019). The class is initiated with a robot description file, updates the robot's configuration, and implements APIs that return the robot's kinematics and dynamics information, such as link Jacobians and the robot's mass matrix.

ControlArchitecture: This is the main class that computes Commands, and it is composed of *StateMachine*, *TaskContactContainer*, and *IHWBC*. *StateMachine* includes a finite number of states, where each one determines distinct planner or controller parameter. State transitions are triggered by predefined temporal parameters, contact events, or user interrupts through the *InterruptLogic* class. At each state in the state machine, the *TrajectoryManager* communicates with the *Planner* and updates the desired task accelerations $\ddot{x}_i^d$ and reaction wrenches f_r^d . The *HierarchyManager* updates the implicit hierarchy between the tasks by modifying the task weights w_i , and the *ReactionForceManager* updates the maximum reaction force $f_r^{\max}$. Note that continuous changes to w_i and $f_r^{\max}$ ensure smooth task and contact transitions. The *TaskContactContainer* contains a list of *Tasks* and *Contacts* that are used in *IHWBC* and which provide APIs for returning the quantities required to solve the QP optimization described in Eq. 5.

PnC provides a predefined task and contact library that can be reused in other WBC implementations thus allowing flexible use for different applications. Currently, four tasks and two contact types are provided in the library: joint position task, Cartesian position task, orientation task, COM task, point contact, and surface contact. *IHWBC* solves a QP optimization and integrates the resulting joint accelerations to compute τ^{cmd} , $\dot{q}^{\text{cmd}}$, and q^{cmd} , which are then sent to the robot using the *Command* class.

5 RESULTS

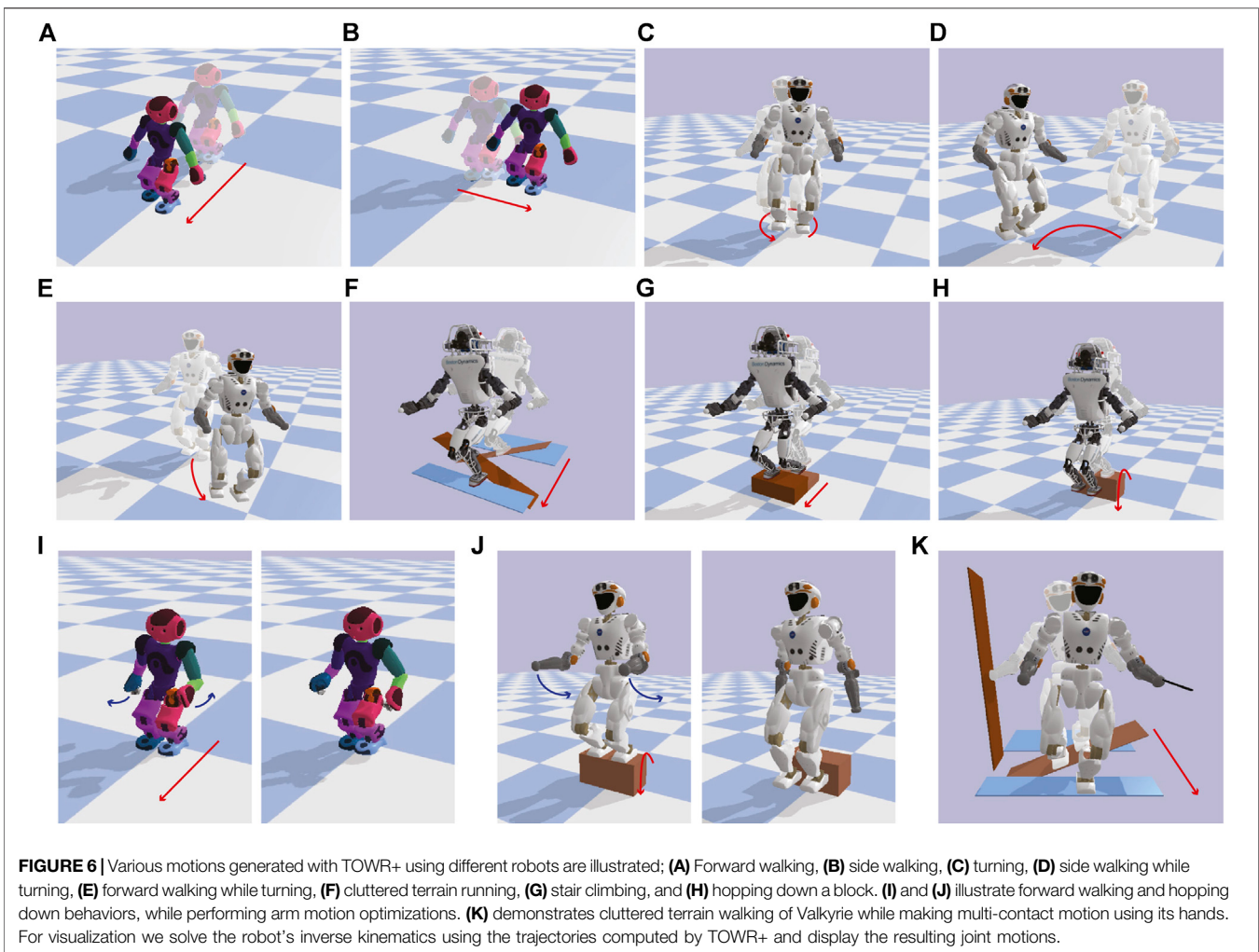
In this section, we evaluate our planning algorithm, TOWR+, and the whole-body control framework, IHWBC, with various robots in both simulation and real hardware. The simulation and experiment videos are available at <https://youtu.be/XionNtDvM20>.

5.1 TOWR+

We demonstrate a variety of motions generated with TOWR+ algorithm. We also study the effectiveness of our CRB model which employs the proposed centroidal inertia network. Finally, we perform a numerical complexity analysis of the TOWR+ algorithm.

5.1.1 Illustrative Examples

Here, we consider various locomotion tasks with the NAO robot from SoftBank (Shamsuddin et al., 2011), Valkyrie from NASA (Radford et al., 2015), and ATLAS from Boston Dynamics. Before solving the trajectory optimization, we first trained a centroidal inertia network for each robot. To create the dataset, we followed the pipeline shown in Figure 4. We considered the robots' kinematic limits and use them to sample a wide range of parameters so that the dataset includes configurations for many types of locomotion. The probability distribution used for sampling can be found in the open-source code repository



we mentioned earlier. We generated 10^4 end-effectors' motions and then collected 15 data points per motion. We built the centroidal inertia network using a multi-layer perceptron with two hidden layers containing 64 nodes and tanh activation functions for differentiability. Finally, we obtained an analytic expression of the trained regressor and its Jacobian with respect to the inputs using the CasADi tool (Andersson et al., 2019).

Equipped with the trained centroidal inertia network, we generated various motions on the NAO, Valkyrie, and ATLAS moving in different terrains, as shown in **Figure 6**. For each example, we specified an initial and final state of the system, total duration of the motion, the number of contacts for each end-effector, and terrain information to generate a desirable walking trajectory. More detailed problem specifications including terrain information can be found in the provided code repository. We then solved for the base motion trajectories, $[r(t), \theta(t)]$, each end-effectors' motions $[p_i(t), \phi_i(t)]$ and wrench trajectories $[f_i(t), \tau_i(t)]$, as well as the time duration for each phase (ΔT_i). As shown in Subfigures 6A to 6H, our TOWR+ motion planner generated various locomotion behaviors including running over a cluttered terrain and jumping over a ground block by optimizing the phase duration vectors. By including the arms in the optimization

process, TOWR+ discovered new arm motions that modify the robot's composite inertia as shown in Subfigures 6I and 6J. For instance, the NAO robot adapted its centroidal inertia using its arms such that the volume of its inertia ellipsoid changed by 13% during forward walking (**Figure 6I**) compared to the case without arm motions (**Figure 6A**). To the best of our knowledge, inertia shaping capabilities using arm motions used for complex terrain locomotion planning have not been achieved by other simplified model-based motion planners. Finally, TOWR+ successfully produced motion trajectories to handle multi-contact scenarios as shown in **Figure 6K**.

For all examples in **Figure 6**, we discretized the base motion trajectories with a 0.1 s timestep. We used two polynomials to represent the foot motions during the swing phase and three polynomials for the contact wrench profiles during the stance phase. We enforced kinematic reachability constraints every 0.08 s and dynamics constraints at every 0.1 s. We imposed the terrain elevation constraint and the contact wrench cone constraint only at the junctions of the feet polynomials, but these constraints could be enforced with a finer granularity. For the stair and the block shown in **Figures 6G,H,J**, we smoothly increased or decreased the height around the edges with a

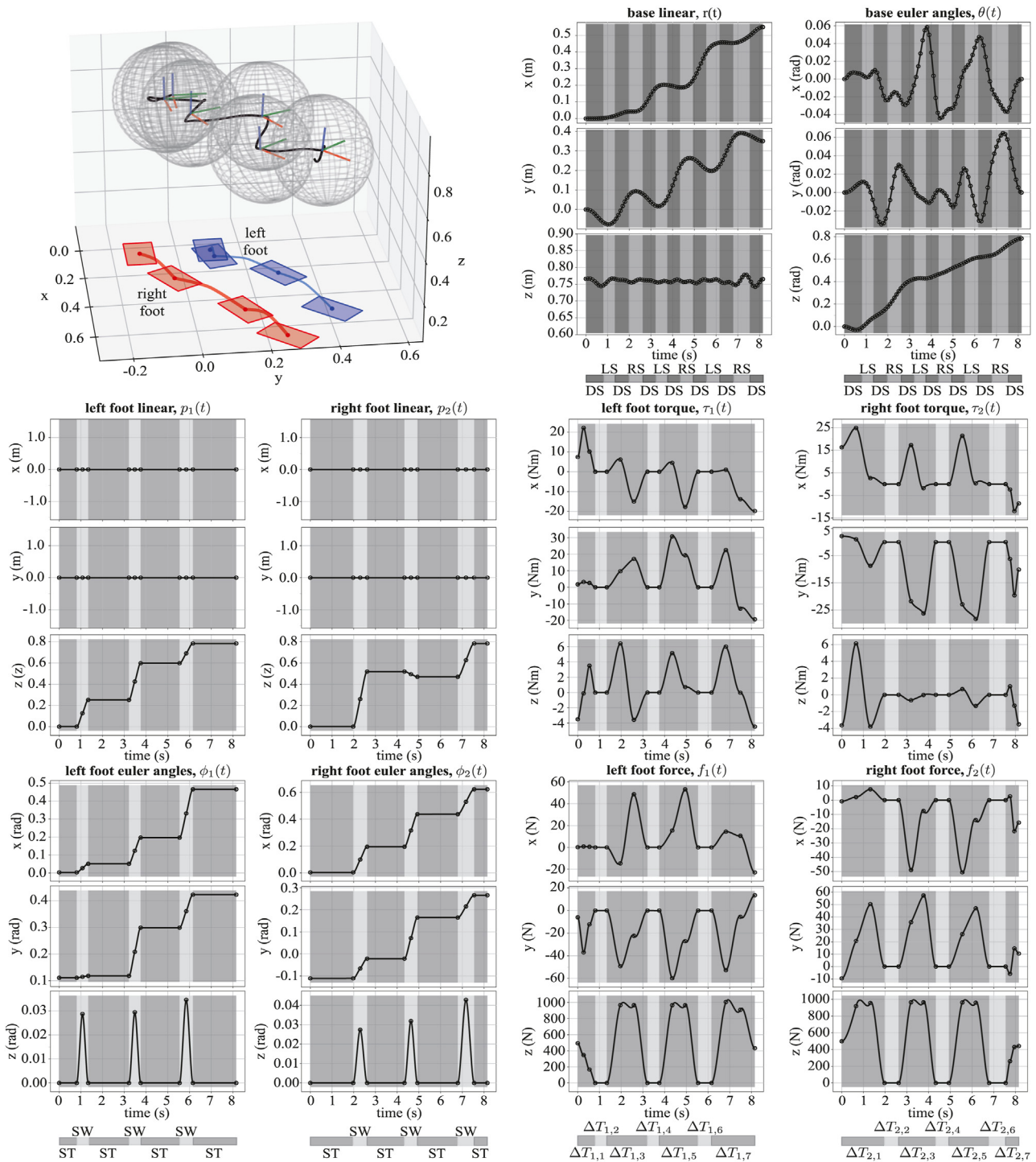


FIGURE 7 | We illustrate the trajectories for Valkyrie walking forward while turning. The upper left figure shows the motions in Cartesian space with varying centroidal inertia ellipsoids.

small margin so that the solver could access the gradient information of the terrain. Finally, we solved the optimization problem using the Interior point method solver, Ipopt (Winkler, 2018).

5.1.2 Quantitative Analysis

First, we study solution trajectories for the example shown in **Figure 6E**. The solution trajectories for this example are shown in

TABLE 2 | Average root-mean-squared percent errors of the centroidal inertia (I_{xx} , I_{yy} , I_{zz}) and angular momentum rate ($\dot{L}$) predictions when using our trained our centroidal inertia network and a constant inertia computed at a nominal pose.

	I_{xx}	I_{yy}	I_{zz}	$\dot{L}$
Centroidal inertia network	1.05 (%)	0.43 (%)	0.114 (%)	0.99 (%)
Constant inertia at nominal pose	10.69 (%)	9.41 (%)	8.05 (%)	8.91 (%)

Figure 7. As mentioned, the optimization nodes for the base motions are equally distributed with a 0.1 s interval, whereas the nodes for the feet motions and wrenches are defined based on the contact phase. For instance, the feet motions in swing mode and the wrench profiles in stance mode are represented with two and three polynomials respectively. At the same time, a single polynomial with a constant value is used to represent the stance phase of the feet. Another polynomial with zero value is used to represent the wrench trajectories during the swing phase. This strategy ensures that the feet do not slip during contact and do not generate wrenches in the air. The gait pattern and schedule are also optimized using the phase duration variables, ΔT_1 and ΔT_2 .

We now evaluate the centroidal inertia network during the walking and turning behavior shown in **Figure 6E**. At every 0.1 s, we solved the inverse kinematics problem using the output trajectories and computed ground truth centroidal inertia tensor and angular momentum rate. Then, we computed the average root-mean-squared percentage errors for the robot's inertia and angular momentum rate when using our network's predictions with respect to ground truth values. For comparison, we also implemented TOWR+ with a constant centroidal inertia tensor computed at a nominal pose instead of using the composite centroidal inertia network, and computed the previous errors again. As shown in **Table 2**, the centroidal inertia network provides much more accurate predictions, which means that the solution trajectories are achieved with higher fidelity.

We also study computational efficiency of our centroidal inertia network by measuring computation time to evaluate our network and its Jacobian during the optimization process. We take the example with Valkyrie shown in **Figure 6E**. The evaluation of the regressor function and its Jacobian took place at 8,216 and 39,840 times during the optimization and took 0.37 and 5.58 s, respectively. To compare this, we solve the robot's inverse kinematics (Buss, 2004) to obtain joint positions and then obtain the centroidal inertia instead of using our pre-trained centroidal inertia. With the inverse kinematics method, the total evaluation time of the centroidal inertia tensor and its Jacobian took 49.72 and 4,302.72 s, respectively. Note that we provided an analytic expression of the regressor network and its Jacobian to the solver, but in the inverse kinematics approach, we had to use the numerical Newton-Raphson method and the finite difference method.

Finally, we perform a numerical complexity analysis of our trajectory optimization framework. As illustrated in **Figure 8**, we defined six different forward walking behaviors with different goal distances, total duration of the motion, and total number of steps. For each case, we counted the number of decision variables, constraints, optimization iterations with respect to the wall-clock

time. As the complexity increases linearly, the number of decision variables and constraints increases linearly too, whereas the wall-clock time to solve the problem increases almost linearly. The number of iterations to find a solution remains the same, regardless of process complexity during forward walking tasks. The numerical computations performed in this section were run on a single core of an Intel I7-4770HQ at 2.5 GHz.

5.2 IHWBC

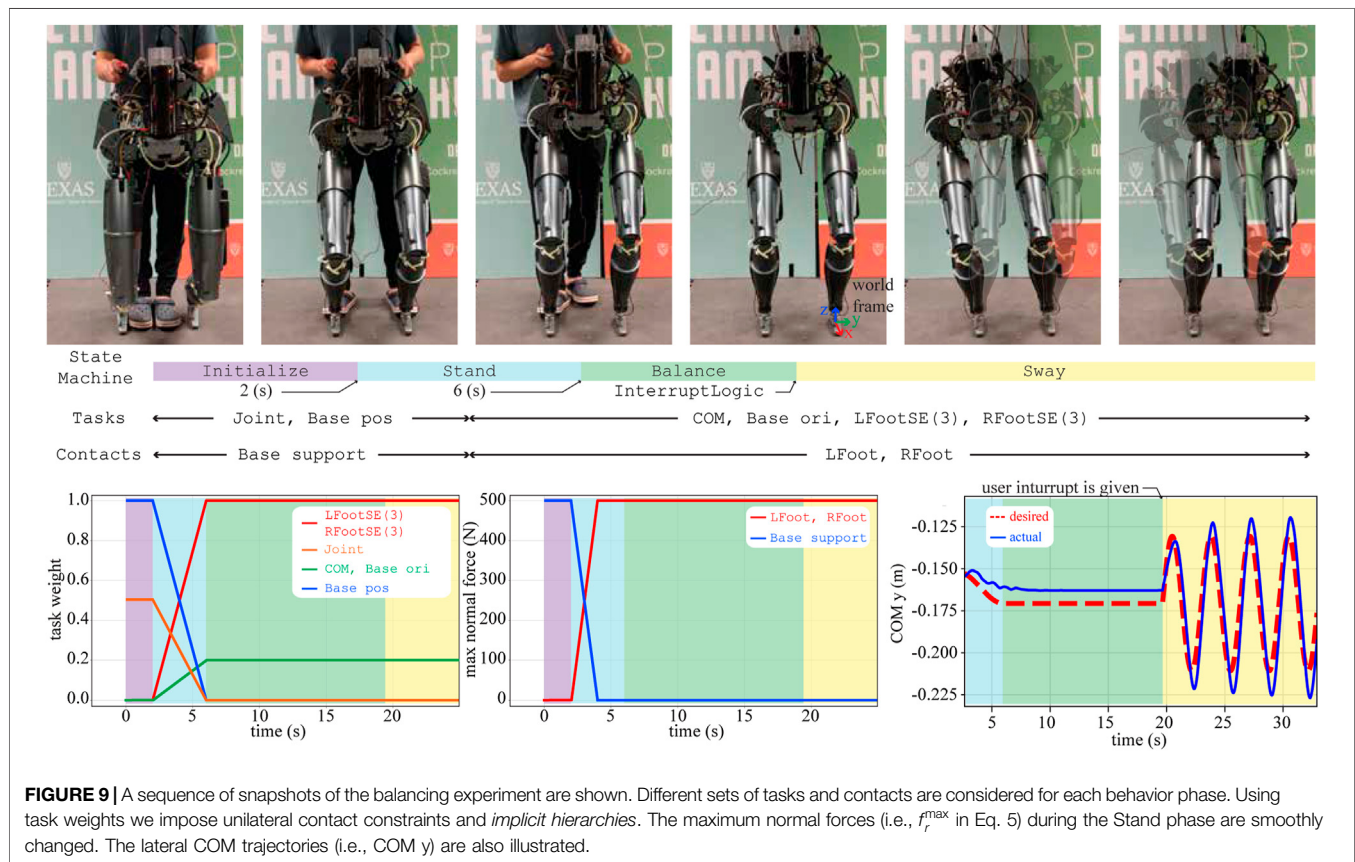
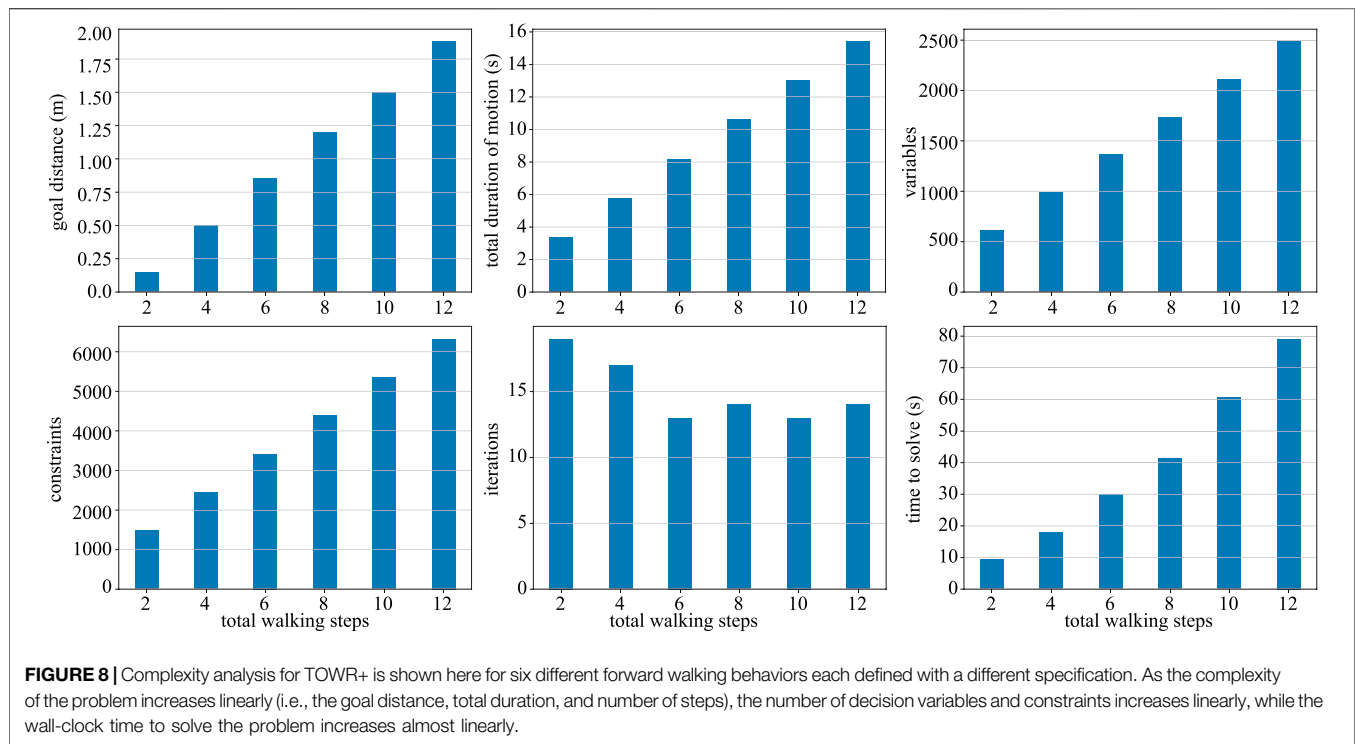
Here, we tested our biped DRACO robot balancing using IHWBC. As shown in **Figure 9**, the StateMachine class that we used consists of a sequence of six different states to achieve the desired balancing behavior.

Initialize: DRACO is initialized in the air where its base (i.e., torso) is supported by an overhead gantry with guidance from the experimenter. During the Initialize phase, DRACO moves its joints to a predefined target configuration for standing by employing reaction forces with the support ground. To be more specific, IHWBC controls the 10-dimensional joint positions and the 3-dimensional base position using a 3-dimensional reaction force from the base support contact. The desired base motions are controlled with zero accelerations with a high task weight to be fixed, while the desired joint motions are controlled using an interpolation between initial and target configurations. The Initialize phase lasts for 2 s and then switches to the Stand phase.

Stand: In this phase, DRACO uses the reaction forces from the ground to lift itself up. During this phase, IHWBC performs a smooth task transition from one task set (i.e., joint task and base position task) to another task set [i.e., 3-dimensional COM task, 3-dimensional base orientation task, 6-dimensional left and right foot SE(3) tasks] as well as a continuous contact transition from one contact set (i.e., base support contact) to another contact set (i.e., 6-dimensional left and right foot contacts⁶). The smooth changes of the task weights and maximum normal force in **Figure 9** ensure continuous task and contact transitions. Next, the desired trajectories for the COM are determined by interpolating between initial and target positions, where the target x and the y position correspond to the middle of the feet and the target height z is set to be 0.7 m. The desired orientation for the base frame is set to be identical to the world frame. The desired accelerations for the feet motion tasks are set to zero to ensure the fulfillment of unilateral contact constraints. The Stand phase lasts for 4 s and switches on to the Balance phase.

Balance: During this phase, DRACO balances without any support from the overhead gantry or experimenter. During this phase, IHWBC maintains the COM and base orientation steady and forces the feet accelerations to become zero to prevent slipping. There are no contact forces from the handles due to experimenter interactions, and the feet control their reaction wrenches for balancing. The Balance phase lasts until receiving a user interrupt *via* InterruptLogic and then switches to the Sway phase.

⁶DRACO's line-foot is approximated using a narrow surface contact.



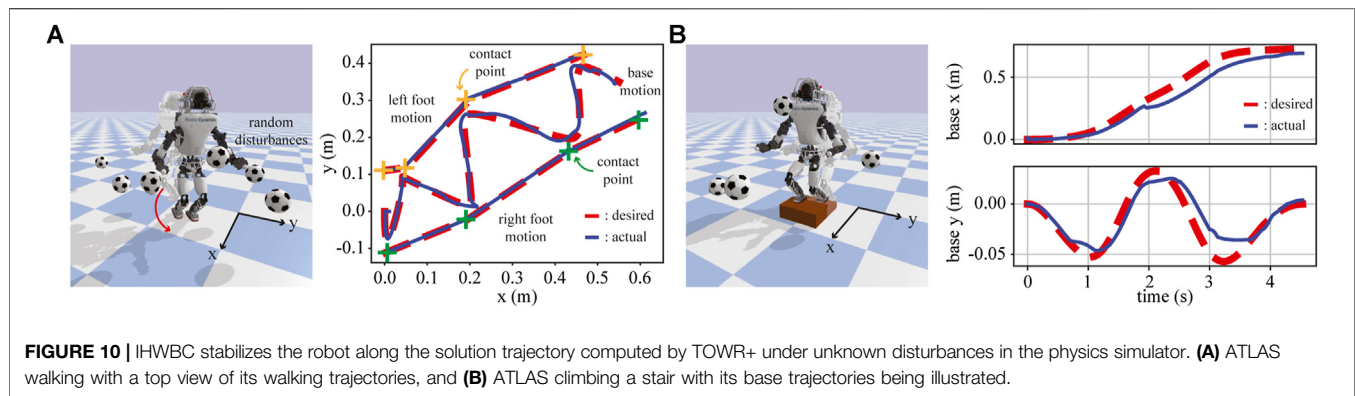


FIGURE 10 | IHWBC stabilizes the robot along the solution trajectory computed by TOWR+ under unknown disturbances in the physics simulator. **(A)** ATLAS walking with a top view of its walking trajectories, and **(B)** ATLAS climbing a stair with its base trajectories being illustrated.

Sway: In this state, the desired lateral motions for the COM task are determined using a sinusoidal trajectory planner using 0.04 m for the amplitude and 0.3 Hz for the frequency. The other tasks and contacts are kept identical to those from the previous Balance phase. The tracking performance in the lateral direction is shown in **Figure 9**.

During this balance experiment, we determined the desired joint accelerations using IHWBC and integrated them to provided the desired joint position and velocity commands to send to the low-level controller of the robot.

5.3 End-to-End Evaluation

Here, we integrate our TOWR+ and IHWBC algorithms and evaluate the full PnC framework both in simulation and in hardware experiments.

5.3.1 ATLAS Walking With Unknown Disturbances

We compute ATLAS walking trajectories using TOWR+ and stabilize the robot using IHWBC in the physics simulator, PyBullet (Coumans and Bai, 2016–2019). As shown in **Figure 10**, we consider a case involving forward walking while turning and another case involving stair climbing. While controlling the robot, we generated random disturbances by throwing a 500 g soccer ball every 1 s. IHWBC rejected successfully the disturbances and achieved robust locomotion behaviors without falling.

5.3.2 DRACO Step-and-Stop Experiment

In this section, we integrate TOWR+ and IHWBC to demonstrate step-and-stop behaviors using the DRACO biped. **Figure 11** shows three consecutive step-and-stop motions. Similarly to the balance experiment, we initialize the robot to the Balance phase. Then, an InterruptLogic (e.g., keyboard stroke) initiates TOWR+ to generate step-and-stop trajectories. Here, the robot swings either its left or right foot and stops at the nominal standing pose for 1.2 s. Once a solution is achieved by TOWR+, phase transitions are triggered based on the phase duration while the robot is stabilized along the trajectory using IHWBC.

ContactTransitionEnd (CTE): In this phase, we take off a small portion (i.e., 0.1 s) of the last part of the double support phase (i.e., Balance phase) and modify it to achieve smooth task and

contact transitions. More specifically, CTE employs feet motion tasks, COM position task, and base orientation task, as well as feet contacts. Here, we continuously reduce the task weight and maximum normal force for each foot about to start swinging so that the robot can smoothly break contact. In this transition phase, the desired accelerations for the swinging foot are still zero, but the unilateral contact constraint becomes weaker as the task weight decreases. Similarly, IHWBC reduces the reaction forces from the swinging foot *via* reducing the maximum normal force $f_r^{\max}$. The task weights and the maximum normal force during this phase are illustrated in **Figure 11**.

Swing: In this phase, the robot swings its leg using the trajectories computed by TOWR+. The phase employs the same type of tasks then CTE but with an implicit priority order as follows: contact foot motion task, swing foot motion task, COM task, and base orientation task. During step-and-stop motions, the desired foot and COM trajectories are shown in **Figure 11**. We also illustrate the desired normal direction of the contact reaction forces computed by IHWBC during swinging.

ContactTransitionStart (CTS): In this phase, we take off a small portion (i.e., 0.1 s) of the double support phase (i.e., the Balance phase) and modify it to achieve smooth task and contact transitions. CTS employs the same tasks and contacts as in the Swing phase but continuously increasing the task weight and maximum normal force of the swinging foot so that the robot can smoothly make contact. The task weights and max normal force during this phase are illustrated in **Figure 11**.

In summary, we have incorporated the proposed TOWR+ and IHWBC algorithms to achieve step-and-stop motions with DRACO. Thanks to the use of the CRB model in TOWR+, we have been able to plan dynamic motion trajectories employing accurate centroidal inertia information compared to other simplified model-based planners, such as the inverted pendulum-based planner. In addition to this, our IHWBC method provides flexibility for multi task management and smooth transitioning capabilities for step-and-stop motions resulting in small foot placement errors (less than 0.036 m as shown in **Figure 11**). To the best of our knowledge, stepping behaviors on a line-foot biped with significant leg mass and centroidal inertia variations has not been achieved before. Although this is an achievement of TOWR+ and IHWBC, we admit that there is a gap between the simulations with other

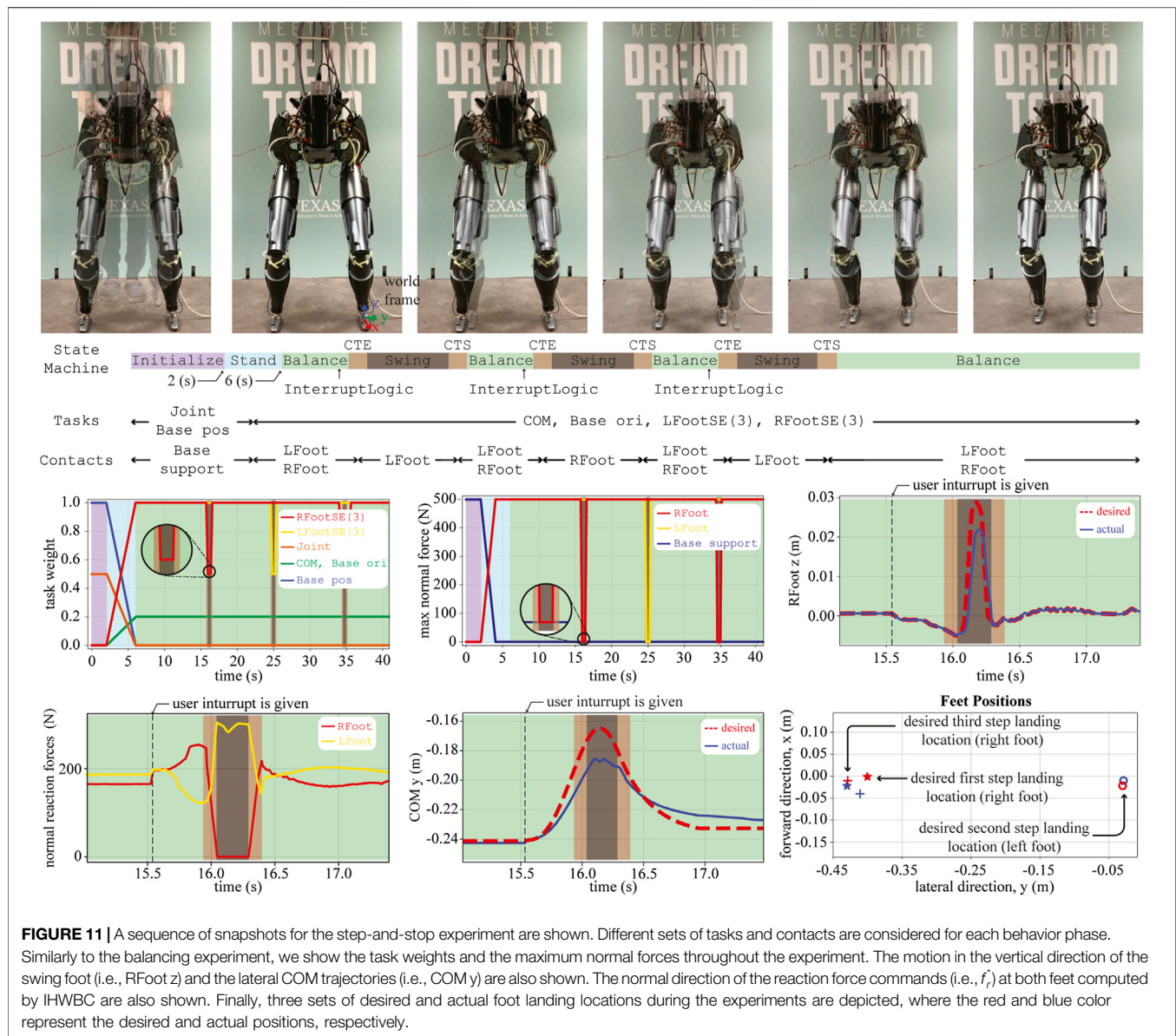


FIGURE 11 | A sequence of snapshots for the step-and-stop experiment are shown. Different sets of tasks and contacts are considered for each behavior phase. Similarly to the balancing experiment, we show the task weights and the maximum normal forces throughout the experiment. The motion in the vertical direction of the swing foot (i.e., RFoot z) and the lateral COM trajectories (i.e., COM y) are also shown. The normal direction of the reaction force commands (i.e., $\hat{r}_i$) at both feet computed by IHWBC are also shown. Finally, three sets of desired and actual foot landing locations during the experiments are depicted, where the red and blue color represent the desired and actual positions, respectively.

robots and the DRACO step-and-stop experiment. In the following section, we discuss hardware limitations that hinder simulation like behaviors and future research directions that can possibly push forward the limits.

6 DISCUSSION

In this paper, we have proposed a trajectory optimization framework, TOWR+, and a whole-body control formulation, IHWBC, plus their implementation in an open-source code repository, PnC. TOWR+ employ a novel composite rigid body inertia model, CRB, to model humanoid robot dynamics more accurately than simpler centroidal dynamic model. In addition, it incorporates extensions to deal with flat feet contacts including 6D reaction wrenches and SE(3) feet

trajectories. Like TOWR, TOWR+ does not rely on integer programming or complementarity constraints.

We have demonstrated various advanced locomotion behaviors on different humanoid robots and conducted a thorough analyses of complexity of TOWR+. For feedback whole-body control, we have devised a new algorithm, IHWBC, which establishes implicit hierarchies between multiple tracking tasks and defines soft constraints to deal with unilateral contact constraint. This method allows for smooth task and contact transitions and provides versatility and ease-of-use for general biped robots and humanoids. Finally, we have integrated the proposed TOWR+ novel trajectory optimization method and IHWBC to demonstrate balancing and a step-and-stop behaviors in our real DRACO biped robot.

Although the step-and-stop experiment has been implemented in the real hardware, there are a few hardware

limitations on Draco that hinder more complex demonstrations. Due to the small line feet and the comparatively heavy distal mass, the dynamic stability of the robot requires quick swing motions (<0.3 s) and a small swing height (<3 cm). However, the physical robot has a motion bandwidth that is upper bounded by the mechanical linkage structures and the actuation power. In the step-and-stop experiment, we were able to control the stepping motion within a 3.6 cm error using IHWBC, which is enough for one step-and-stop motion but not for continuous walking.

With that in mind, we consider a few future research directions. Although TOWR+ is computationally efficient in discovering novel contact sequences and generating locomotion trajectories, it is not fast enough to be implemented as a receding horizon controller, which is necessary for the robust continuous walking. We are interested in devising generative models to provide a good initial guess for warm starting TOWR+. Finally, we are looking forward to deploying the proposed end-to-end framework in our upcoming humanoid robot which has biped legs, with 6 DOF for each of them, and a two arm manipulation torso with an articulated head equipped with a vision sensor.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: <https://github.com/junhyeokahn/PyPnC>.

REFERENCES

- Ahn, J., Campbell, O., Kim, D., and Sentis, L. (2018). "Fast Kinodynamic Bipedal Locomotion Planning with Moving Obstacles," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. Madrid, Spain: IROS, 177–184. doi:10.1109/IROS.2018.8594156
- Ahn, J., Kim, D., Bang, S., Paine, N., and Sentis, L. (2019). "Control of a High Performance Bipedal Robot Using Viscoelastic Liquid Cooled Actuators," in 2019 IEEE-RAS 19th International Conference on Humanoid Robots. Toronto, ON: Humanoids, 146–153. doi:10.1109/Humanoids43949.2019.9035023
- Ahn, J., Lee, J., and Sentis, L. (2020). Data-efficient and Safe Learning for Humanoid Locomotion Aided by a Dynamic Balancing Model. *IEEE Robot. Autom. Lett.* 5, 4376–4383. doi:10.1109/LRA.2020.2990743
- Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). Casadi: a Software Framework for Nonlinear Optimization and Optimal Control. *Math. Prog. Comp.* 11, 1–36. doi:10.1007/s12532-018-0139-4
- Apgar, T., Clary, P., Green, K., Fern, A., and Hurst, J. (2018). "Fast Online Trajectory Optimization for the Bipedal Robot Cassie," in *Robotics: Science and Systems* (Pittsburgh, Pennsylvania, USA, 101, 14.
- Bolotnikova, A., Gergondet, P., Tanguy, A., Courtois, S., and Kheddar, A. (2020). "Task-space Control Interface for Softbank Humanoid Robots and its Human-Robot Interaction Applications," in 2021 IEEE/SICE International Symposium on System Integration. Iwaki, Fukushima: SII. [Dataset]. doi:10.15607/rss.2018.xiv.054
- Buss, S. (2004). Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares Methods. *IEEE Trans. Robotics Automation* 17.
- Caron, S., Escande, A., Lanari, L., and Mallein, B. (2020). Capturability-based Pattern Generation for Walking with Variable Height. *IEEE Trans. Robot.* 36, 517–536. doi:10.1109/TRO.2019.2923971

AUTHOR CONTRIBUTIONS

All authors contributed to the theory and implementation of the study. To be more specific, JA formulated and implemented TOWR+, and conducted the simulations and hardware experiments in the article. SJ formulated and implemented IHWBC, and validated it using simulations and hardware experiments. SB contributed to the data generation pipeline for the centroidal inertia network training and the software maintenance. LS supervised the research and revised and approved the submitted manuscript.

FUNDING

This work was supported by the Office of Naval Research, ONR Grant #N000141512507.

ACKNOWLEDGMENTS

The authors would like to thank the members of the Human Centered Robotics Laboratory at the University of Texas at Austin for their great help and support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2021.712239/full#supplementary-material>

- Caron, S., Pham, Q.-C., and Nakamura, Y. (2015). "Stability of Surface Contacts for Humanoid Robots: Closed-form Formulae of the Contact Wrench Cone for Rectangular Support Areas," in 2015 IEEE International Conference on Robotics and Automation. Seattle, WA: ICRA, 5107–5112. doi:10.1109/ICRA.2015.7139910
- Carpentier, J., and Mansard, N. (2018). Multicontact Locomotion of Legged Robots. *IEEE Trans. Robot.* 34, 1441–1460. doi:10.1109/TRO.2018.2862902
- Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamirault, F., Stasse, O., et al. (2019). "The Pinocchio C++ Library: A Fast and Flexible Implementation of Rigid Body Dynamics Algorithms and Their Analytical Derivatives," in 2019 IEEE/SICE International Symposium on System Integration. Paris, France: SII, 614–619. doi:10.1109/SII.2019.8700380
- Coumans, E., and Bai, Y. (2016–2019). Pybullet, a python Module for Physics Simulation for Games, Robotics and Machine Learning. Available at: <http://pybullet.org> [Dataset].
- Dai, H., Valenzuela, A., and Tedrake, R. (2014). "Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics," in 2014 IEEE-RAS International Conference on Humanoid Robots (Madrid, Spain), 295–302. doi:10.1109/HUMANOIDS.2014.7041375
- Del Prete, A. (2018). Joint Position and Velocity Bounds in Discrete-Time Acceleration/torque Control of Robot Manipulators. *IEEE Robot. Autom. Lett.* 3, 281–288. doi:10.1109/LRA.2017.2738321
- Del Prete, A., Mansard, N., Ramos, O. E., Stasse, O., and Nori, F. (2016). Implementing Torque Control with High-Ratio Gear Boxes and without Joint-Torque Sensors. *Int. J. Hum. Robot.* 13, 1550044.
- Engelsberger, J., Ott, C., and Albu-Schäffer, A. (2015). Three-dimensional Bipedal Walking Control Based on Divergent Component of Motion. *IEEE Trans. Robot.* 31, 355–368. doi:10.1109/TRO.2015.2405592

- Faraji, S., Razavi, H., and Ijspeert, A. J. (2019). Bipedal Walking and Push Recovery with a Stepping Strategy Based on Time-Projection Control. *Int. J. Robotics Res.* 38, 587–611. doi:10.1177/0278364919835606
- Feng, S., Whitman, E., Xinjilefu, X., and Atkeson, C. G. (2014). “Optimization Based Full Body Control for the Atlas Robot,” In 2014 IEEE-RAS International Conference on Humanoid Robots (Madrid, Spain), 120–127. doi:10.1109/HUMANOID.2014.7041347
- Goldfarb, D., and Idnani, A. (1983). A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs. *Math. Programming* 27, 1–33. doi:10.1007/BF0259196210.1007/bf02591962
- Grimminger, F., Meduri, A., Khadiv, M., Viereck, J., Wüthrich, M., Naveau, M., et al. (2020). An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research. *IEEE Robot. Autom. Lett.* 5, 3650–3657. doi:10.1109/Lra.2020.2976639
- Hereid, A., Cousineau, E. A., Hubicki, C. M., and Ames, A. D. (2016). “3d Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics,” in 2016 IEEE International Conference on Robotics and Automation. Stockholm, Sweden: ICRA, 1447–1454. doi:10.1109/ICRA.2016.7487279
- Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., and Righetti, L. (2016). Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid. *Auton. Robot.* 40, 473–491. doi:10.1007/s10514-015-9476-6
- IHMCRobotics (2018). Ihmc-Open-Robotics-Software. Available at: <https://github.com/ihmcrobotics/ihmc-open-robotics-software>. [Dataset].
- Jorgensen, S. J. (2020). *Towards Deploying Legged Humanoids in Human Environments*. TX, USA: Ph.D. thesisThe University of Texas at Austin.
- Jorgensen, S. J., Vedantam, M., Gupta, R., Cappel, H., and Sentis, L. (2020). “Finding Locomanipulation Plans Quickly in the Locomotion Constrained Manifold,” in 2020 IEEE International Conference on Robotics and Automation. Paris, France: ICRA, 6611–6617. doi:10.1109/ICRA40945.2020.9197533
- Kim, D., Carlo, J. D., Katz, B., Bledt, G., and Kim, S. (2019). *Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control*, 1909. CoRR abs/, 06586.
- Kim, D., Jorgensen, S. J., Lee, J., Ahn, J., Luo, J., and Sentis, L. (2020). Dynamic Locomotion for Passive-Ankle Biped Robots and Humanoids Using Whole-Body Locomotion Control. *Int. J. Robotics Res.* 39, 936–956. doi:10.1177/0278364920918014
- Kim, D., Lee, J., Ahn, J., Campbell, O., Hwang, H., and Sentis, L. (2018). “Computationally-robust and Efficient Prioritized Whole-Body Controller with Contact Constraints,” in 2018 IEEE/RIS International Conference on Intelligent Robots and Systems. Madrid, Spain: IROS, 1–8. doi:10.1109/IROS.2018.8593767
- Kim, S., Jang, K., Park, S., Lee, Y., Lee, S. Y., and Park, J. (2019). Continuous Task Transition Approach for Robot Controller Based on Hierarchical Quadratic Programming. *IEEE Robot. Autom. Lett.* 4, 1603–1610. doi:10.1109/LRA.2019.2896769
- Koolen, T., Bertrand, S., Thomas, G., de Boer, T., Wu, T., Smith, J., et al. (2016). Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas. *Int. J. Hum. Robot.* 13, 1650007. doi:10.1142/S0219843616500079
- Koolen, T., de Boer, T., Rebula, J., Goswami, A., and Pratt, J. (2012). Capturability-based Analysis and Control of Legged Locomotion, Part 1: Theory and Application to Three Simple Gait Models. *Int. J. Robotics Res.* 31, 1094–1113. doi:10.1177/0278364912452673
- Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., et al. (2016). Optimization-based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot. *Auton. Robot.* 40, 429–455. doi:10.1007/s10514-015-9479-3
- Lee, J., Mansard, N., and Park, J. (2012). Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control. *IEEE Trans. Robot.* 28, 1260–1277. doi:10.1109/TRO.2012.2210293
- Lee, J., X. Grey, M., Ha, S., Kunz, T., Jain, S., Ye, Y., et al. (2018). Dart: Dynamic Animation and Robotics Toolkit. *Joss* 3, 500. doi:10.21105/joss.00500
- Mastalli, C., Budhiraja, R., Merkt, W., Saurel, G., Hammoud, B., Naveau, M., et al. (2020). “Crocodyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control,” in 2020 IEEE International Conference on Robotics and Automation. Paris, France: ICRA, 2536–2542. doi:10.1109/ICRA40945.2020.9196673
- Mordatch, I., Todorov, E., and Popović, Z. (2012). Discovery of Complex Behaviors through Contact-Invariant Optimization. *ACM Trans. Graph.* 31. doi:10.1145/2185520.2185539
- Park, J., and Khatib, O. (2008). Robot Multiple Contact Control. *Robotica* 26, 667–677. doi:10.1017/S0263574708004281
- Ponton, B., Herzog, A., Del Prete, A., Schaal, S., and Righetti, L. (2018). “On Time Optimization of Centroidal Momentum Dynamics,” in 2018 IEEE International Conference on Robotics and Automation. Brisbane, Australia. ICRAIEEE, 5776–5782. doi:10.1109/icra.2018.8460537
- Ponton, B., Khadiv, M., Meduri, A., and Righetti, L. (2021). Efficient Multicontact Pattern Generation with Sequential Convex Approximations of the Centroidal Dynamics. *IEEE Trans. Robot.*, 1–19. doi:10.1109/TRO.2020.3048125
- Posa, M., Cantu, C., and Tedrake, R. (2013). A Direct Method for Trajectory Optimization of Rigid Bodies through Contact. *Int. J. Robotics Res.* 33, 69–81. doi:10.1177/0278364913506757
- Radford, N. A., Strawser, P., Hambuchen, K., Mehling, J. S., Verdeyen, W. K., Donnan, A. S., et al. (2015). Valkyrie: NASA’s First Bipedal Humanoid Robot. *J. Field Robotics* 32, 397–419. doi:10.1002/rob.21560
- Rocchi, A., Hoffman, E. M., Caldwell, D. G., and Tsagarakis, N. G. (2015). “Opensot: A Whole-Body Control Library for the Compliant Humanoid Robot Coman,” in 2015 IEEE International Conference on Robotics and Automation. Seattle, WA: ICRA, 6248–6253. doi:10.1109/ICRA.2015.7140076
- Saab, L., Ramos, O. E., Keith, F., Mansard, N., Souères, P., and Fourquet, J.-Y. (2013). Dynamic Whole-Body Motion Generation under Rigid Contacts and Other Unilateral Constraints. *IEEE Trans. Robot.* 29, 346–362. doi:10.1109/TRO.2012.2234351
- Sentis, L., and Khatib, O. (2005). Synthesis of Whole-Body Behaviors through Hierarchical Control of Behavioral Primitives. *Int. J. Hum. Robot.* 02, 505–518. doi:10.1142/S0219843605000594
- Seyde, T., Shrivastava, A., Engelsberger, J., Bertrand, S., Pratt, J., and Griffin, R. J. (2018). “Inclusion of Angular Momentum during Planning for Capture point Based Walking,” in 2018 IEEE International Conference on Robotics and Automation. Brisbane, QLD, Australia. ICRA, 1791–1798. doi:10.1109/ICRA.2018.8461140
- Shamsuddin, S., Ismail, L. I., Yusoff, H., Ismarrubie Zahari, N., Bahari, S., Hashim, H., et al. (2011). “Humanoid Robot Nao: Review of Control and Motion Exploration,” in 2011 IEEE International Conference on Control System, Computing and Engineering. Penang, Malaysia, 511–516. doi:10.1109/ICCSC.2011.6190579
- Turlach, B. A., and Weingessel, A. (2011). *Quadprog: Functions to Solve Quadratic Programming Problems*.
- Wiedebach, G., Bertrand, S., Wu, T., Fiorio, L., McCrory, S., Griffin, R., et al. (2016). “Walking on Partial Footholds Including Line Contacts with the Humanoid Robot Atlas,” in 2016 IEEE-RAS 16th International Conference on Humanoid Robots. Cancun, Mexico: Humanoids, 1312–1319. doi:10.1109/HUMANOID.2016.7803439
- Winkler, A. W., Bellicoso, C. D., Hutter, M., and Buchli, J. (2018). Gait and Trajectory Optimization for Legged Systems through Phase-Based End-Effector Parameterization. *IEEE Robot. Autom. Lett.* 3, 1560–1567. doi:10.1109/LRA.2018.2798285
- Winkler, A. W. (2018). *Ifopt - A Modern, Light-Weight, Eigen-Based C++ Interface to Nonlinear Programming Solvers Ipopt and Snopt*. [Dataset]. doi:10.5281/zenodo.1135046

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling Editor declared a past co-authorship with one of the authors LS.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Ahn, Jorgensen, Bang and Sentis. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.