Work those Arms: Toward Dynamic and Stable Humanoid Walking that Optimizes Full-Body Motion

Christian M. Hubicki[1], Ayonga Hereid[1], Michael X. Grey[2], Andrea L. Thomaz[3], and Aaron D. Ames[4]

***Abstract*— Humanoid robots are designed with dozens of actuated joints to suit a variety of tasks, but walking controllers rarely make the best use of all of this freedom. We present a framework for maximizing the use of the full humanoid body for the purpose of stable dynamic locomotion, which requires no restriction to a planning template (e.g. LIPM). Using a hybrid zero dynamics (HZD) framework, this approach optimizes a set of outputs which provides requirements for the motion for all actuated links, including arms. These output equations are then rapidly solved by a whole-body inverse-kinematic (IK) solver, providing a set of joint trajectories to the robot. We apply this procedure to a simulation of the humanoid robot, DRC-HUBO, which has over 27 actuators. As a consequence, the resulting gaits swing their arms, not by a user defining swinging motions *a priori* or superimposing them on gaits *post hoc*, but as an emergent behavior from optimizing the dynamic gait. We also present preliminary dynamic walking experiments with DRC-HUBO in hardware, thereby building a case that hybrid zero dynamics as augmented by inverse kinematics (HZD+IK) is becoming a viable approach for controlling the full complexity of humanoid locomotion.**

I. Introduction

Achieving dynamic walking on humanoids is hard; their dynamics are inherently nonlinear and their numerous actuators render their computational search spaces very high-dimensional. As such, it's often simplest to just ignore their arms when generating their locomotion patterns. This can be a missed opportunity. Beyond the obvious manipulation tasks, arms can be helpful for improving the balance [26] and economy [7] of locomotion by swinging them as part of a dynamic gait. As such, we seek a method of optimizing stable humanoid control that can leverage all of the robot's joints during locomotion tasks.

Researchers who develop humanoid control have achieved varied combinations of full-body freedom and certifications of stability. Since the origin of humanoid control [34], Zero Moment Point (ZMP) approaches have certified balance by ensuring that the center of pressure always rests within the support polygon of its feet [33]. Ensuring this condition has typically required making and/or enforcing linear inverted pendulum model (LIPM) assumptions about the robot dynamics [17], [25], [27], [31]. More recently, robots have begun planning center-of-mass trajectories and contact forces that can reason more general motion for the lumped CoM [9], [10], [19], [20]. In such methods, upper body motions are often generated sequentially, significantly decoupling them from locomotion generation and potentially divorcing the robot from its prior formal stability guarantees. Recent experimental methods have emerged that can plan full-body humanoid motions in an integrated fashion, as a result they can better recognize the merits of dynamic arm motions [8], [11], [15], [18], [22]. We seek a method that can not only plan these motions, but also produce a controller that certifies its dynamic stability.

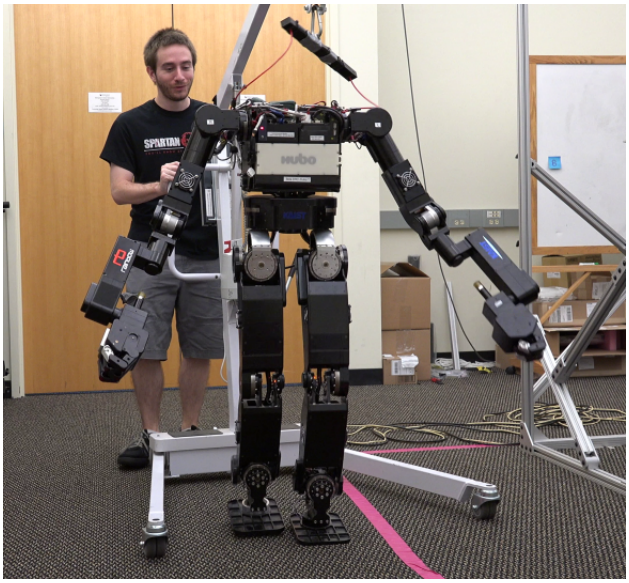

Fig. 1: DRC-HUBO, a humanoid robotic platform with 27 actuated degrees of freedom from its feet to its wrists. This work presents an framework that can generate dynamic locomotion that reasons about the multibody dynamics of all these degrees of freedom, including its swinging arms, without restricting motions to a planning template.

The methodology presented in this paper takes a *hybrid zero dynamics* (HZD) approach to locomotion control [4], [12], [36] in an effort to optimize control and certify stability for the full humanoid dynamics. Zero dynamics offers a natural basis for ensuring stability conditions for controlled nonlinear dynamics, which operates by defining a set of *outputs* and driving those outputs to zero via feedback control. In practice, HZD methods have suffered from scalability issues. When robots have more degrees of freedom (*e.g.* arms), it becomes difficult to optimize these outputs, and

[1]Christian M. Hubicki and Ayonga Hereid are with the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA

[2]Michael Grey is with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30332 USA

[3]Andrea L. Thomaz is with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, 30332 USA

[4]Aaron D. Ames is with the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA

nontrivial to solve nonlinear outputs in real-time.

In this paper, we present a novel framework for humanoid control with supporting simulations and preliminary experimental results for the humanoid robot, DRC-HUBO, walking with dynamic arm swing. This framework employs a large-scale optimization approach to generating stable HZD, and realizes the outputs with whole-body inverse kinematics (IK) methods. This allows us to generate and control a set of output equations that yield a dynamic and stable gait for a whole humanoid model with 27 actuators. Importantly, the humanoid swings its arms as a consequence of optimizing the dynamic gait for energy-efficient locomotion subject to no-net-moment constraints, not by *a priori* specification.

Using a whole body approach to inverse kinematics, we are also able to solve nonlinear outputs at real-time speeds, allowing us to define outputs that are "task-relevant." As a result, the inverse-kinematics solvers can both render the dynamic gait while accommodating any intuitive adjustments requested by a user (e.g. adjust foot placement). We demonstrate these combined methods (HZD+IK) with simulations of stable and dynamic humanoid walking and preliminary experiments with the humanoid robot, DRC-HUBO (Fig. 1).

This paper is structured as follows. Section II presents some preliminaries on the HZD control framework and its application to legged locomotion. Section III discusses our large-scale optimization approach for designing dynamic gaits and the outputs that define them. Section IV explains our methods for solving whole-body kinematics, which solves our output equations at real-time speeds. Section V describes the application of the method to DRC-HUBO, a humanoid robot, presenting simulation and preliminary experimental validation of the framework. We conclude our work in Section VI, summarizing the capabilities of the HZD+IK framework for enabling dynamic humanoid locomotion that really works the arms.

II. Hybrid Zero Dynamics

The approach presented generates stable humanoid controllers by building upon a *hybrid zero dynamics* based theoretical foundation. More precisely, the plastic impacts present in our humanoid model drove us to use a *partial* hybrid zero dynamics formulation [3], which allows for reasonable joint acceleration demands while maintaining stability assurances. We model the locomotion of a bipedal robot as a hybrid system model (see [2], [36] for a formal definition). Given a robot's generalized coordinates, $q \in \mathcal{Q} \subset \mathbb{R}^n$, the continuous dynamics of the system is determined by both the Lagrangian of the robot and holonomic constraints $\eta(q)$, and can be given by the Euler-Lagrange equations [23]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J^T(q)F, \qquad (1)$$

where F is a vector of constraint wrenches, which can be obtained by enforcing the holonomic constraints of the system to be constant. In other words, the second-order derivatives of $\eta(q)$ should be zero:

$$J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} = 0, \qquad (2)$$

where $J(q)$ is the Jacobian matrix of $\eta(q)$, i.e., $J(q) = \frac{\partial \eta}{\partial q}$. The domain of continuous dynamics is determined by the admissible constraints on the robot configuration and constraint wrenches, whereas the discrete event is triggered when the guard condition, $H(q)$, crosses zero. In the specific context of humanoid locomotion, this guard condition will ultimately be used to define ground contact. Formally, we define the switching surface, S, that defines the guard as

$$S = \{(q,\dot{q}) \in T\mathcal{Q} \mid H(q,\dot{q}) = 0, \;\; \dot{H}(q,\dot{q}) < 0\}. \qquad (3)$$

For legged systems, discrete events, such as the impacts of rigid legs colliding with the ground, are assumed to be plastic impacts. Following the presentation in [13], configurations of the system are invariant through an impact, i.e., given pre-impact states $(q^-, \dot{q}^-)$, post-impact configuration yields $q^+ = q^-$, but post-impact velocities, $\dot{q}^+$, need to satisfy the plastic impact equation:

$$\begin{bmatrix} D(q^-) & -J^T(q^-) \\ J(q^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ \delta F \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix}, \qquad (4)$$

where δF are impulsive contact wrenches. This defines the reset map of the system, $(q^+, \dot{q}^+) = \Delta(q^-, \dot{q}^-)$.

To design a walking pattern, one can define a set of *outputs*, y_1 and y_2, which modulate the behavior of a system in order to achieve certain desired trajectories [2]:

$$y_1(q,\dot{q}) = \dot{y}_1^a(q,\dot{q}) - y_1^d(\alpha), \qquad (5)$$

$$y_2(q) = y_2^a(q) - y_2^d(\tau(q),\alpha), \qquad (6)$$

where y_1 is a velocity-modulating output and y_2 is a position-modulating output, and are relative degree 1 and (vector) relative degree 2 respectively by definition (see [29] for the definition of relative degree). Further, y_1^a and y_2^a are the actual outputs and y_1^d and y_2^d are the desired outputs for relative degrees 1 and 2 respectively. If a feedback control law drives these outputs to zero, i.e., $y = (y_1, y_2) \to 0$, it renders the zero dynamics submanifold invariant across all continuous domains. In this paper, we apply the feedback controllers introduced in [2], which yield a set of stable linear output dynamics of the form:

$$\dot{y}_1 = -\varepsilon y_1, \qquad (7)$$

$$\ddot{y}_2 = -2\varepsilon\dot{y}_2 - \varepsilon^2 y_2, \qquad (8)$$

with $\varepsilon > 0$. However, due to the presence of rigid impacts, realizing invariance at impact of all outputs would be very difficult. Therefore, if there exists a set of parameters α such that the invariance of relative degree 2 outputs is guaranteed, then we call the resulting submanifold a *partial hybrid zero dynamics* (PHZD) surface (see [3]) as given by

$$\mathcal{PZ}_\alpha \;=\; \{(q,\dot{q}) \in T\mathcal{Q} \mid y_2(q) = 0,\; \dot{y}_2(q,\dot{q}) = 0\}. \qquad (9)$$

The goal is to ensure that the partial hybrid zero dynamics surface, $\mathcal{PZ}_\alpha$, is invariant through impact, as represented by the partial hybrid zero dynamics condition:

$$\Delta(S \cap \mathcal{PZ}_\alpha) \subset \mathcal{PZ}_\alpha \qquad \text{(PHZD)}$$

or, in other words, $\Delta(q,\dot{q}) \in \mathcal{PZ}_\alpha$ for all $(q,\dot{q}) \in S \cap \mathcal{PZ}_\alpha$.

Given PHZD conditions, we can ensure dynamic stability of a fully actuated bipedal gait provided we can find an appropriate set of outputs. Put more relevantly, *PHZD conditions are sufficient to establish stability when a system is fully actuated* (see Theorem 2 of [2]), as is the case for DRC-HUBO. Finding the parameters, α, that shape these outputs, however, is computationally tricky. For this, we employ a sparse PHZD optimization formulation and large-scale NLP solvers.

III. Optimization

In this section, we briefly present the general form of our large-scale optimization approach to defining outputs, specifically the parameters α that shape them. For computational simplicity, we formulate our desired position-modulating output trajectories[1] using Bézier polynomials of degree M, determined by $M+1$ coefficients:

$$y_2^d(\tau, \alpha[k]) := \sum_{k=0}^{M} \alpha[k] \frac{M!}{k!(M-k)!} \tau^k (1-\tau)^{M-k}, \quad (10)$$

where $\alpha[k]$ is a vector of Bézier polynomial coefficients of each output, and τ is the state-based parameterization of time [3]. These $\alpha[k]$ are each assigned a design variable in a nonlinear optimization seeking to satisfy PHZD conditions and any other task-related constraints and objectives. While we are ultimately interested in only these coefficients, optimization is typically more reliable with formulations that add far more collocated design variables. Particularly, we use a *direct collocation* formulation [32] of the PHZD optimization problem.

We begin by defining a sequence of $N+1$ discrete nodes along the time span of the trajectory:

$$0 = t_0 < t_1 < t_2 < \cdots < t_N = T_I, \quad (11)$$

which forms the basis of our discrete representation of the continuous domains.

Direct collocation formulation methods operate by approximating the evolution of all continuous dynamics. In this case, we use a piecewise Hermite interpolation polynomial [28]. The slope of each polynomial is evaluated at the center point, *i.e.* a *collocation point*, and compared to the closed-form first order dynamics. By driving this difference, or *defect*, to zero we implicitly enforce the dynamics of the system as equality constraints [6], [14]. Let $x = (q, \dot{q})$ be the system states, these constraints can be stated as:

$$x^i - \frac{1}{2}(x^{i+1} + x^{i-1}) - \frac{1}{8}\Delta t_i (\dot{x}^{i-1} - \dot{x}^{i+1}) = 0, \quad \text{(C1)}$$

$$x^{i+1} - x^{i-1} - \frac{1}{6}\Delta t^i (\dot{x}^{i-1} + 4\dot{x}^i + \dot{x}^{i+1}) = 0, \quad \text{(C2)}$$

with $\Delta t_i = t_{i+1} - t_{i-1}$, for $i \in \{1, 3, 5, \ldots, N-1\}$. This form of dynamic defect constraint is applied to all of ordinary differential equation relations between derivatives of design variables (e.g. q and $\dot{q}$). We now define a vector of design variables, $z^i = (T_I^i, q^i, \dot{q}^i, \ddot{q}^i, u^i, F^i, \alpha^i)$, at each node $i \in \{0, 1, 2, \ldots, N\}$, which will ultimately be designed by an optimizer.

To accommodate our continuous domains, we impose defect constraints for the Euler-Lagrange equations and associated holonomic constraints:

$$D(q^i)\ddot{q}^i + C(q^i, \dot{q}^i)\dot{q}^i + G(q^i) - Bu^i - J^T(q^i)F^i = 0, \quad \text{(C3)}$$

$$J(q^i)\ddot{q}^i + \dot{J}(q^i, \dot{q}^i)\dot{q}^i = 0, \quad \text{(C4)}$$

for $i \in \{0, 1, 2, \ldots, N\}$. In addition to dynamics constraints, we formulate admissible continuous domain constraints in terms of a vector, $A(q, F)$, and enforce at each node[2]:

$$A(q^i, F^i) \geq 0. \quad \text{(C5)}$$

Similarly, we enforce discrete dynamics via guard conditions, $H(q)$, at the last node of each continuous domain:

$$H(q^N) = 0, \quad \text{(C6)}$$

$$\dot{H}(q^N, \dot{q}^N) < 0. \quad \text{(C7)}$$

To tie the optimization problem to the output parameters α, we constrain the generalized coordinates to the output dynamics as per (7)-(8) via:

$$\dot{y}_1(q^i, \dot{q}^i, \ddot{q}^i, \alpha^i) + \varepsilon y_1(q^i, \dot{q}^i, \alpha^i) = 0, \quad \text{(C8)}$$

$$\ddot{y}_2(q^i, \dot{q}^i, \ddot{q}^i, \alpha^i) + 2\varepsilon \dot{y}_2(q^i, \dot{q}^i, \alpha^i) + \varepsilon^2 y_2(q^i, \alpha^i) = 0, \quad \text{(C9)}$$

with associated initial conditions

$$y_2(q^0, \alpha^0) = 0, \quad \text{(C10)}$$

$$\dot{y}_2(q^0, \dot{q}^0, \alpha^0) = 0, \quad \text{(C11)}$$

constrained to zero, so that the optimized parameters satisfy the partial hybrid zero dynamics (PHZD) condition. Moreover, to obtain a periodic gait, an equality boundary constraint is enforced in terms of system states at the first and the last node via the plastic impact equation:

$$\mathcal{R}q^0 - q^N = 0, \quad \text{(C12)}$$

$$J(q^N)\mathcal{R}\dot{q}^0 = 0, \quad \text{(C13)}$$

$$D(q^N)(\mathcal{R}\dot{q}^0 - \dot{q}^N) - J^T(q^N)\delta F^N = 0, \quad \text{(C14)}$$

where $\mathcal{R}$ is the relabeling matrix [4].

Finally, we demand that the constant parameters be consistent in all nodes:

$$\alpha^i - \alpha^{i+1} = 0, \quad \text{(C15)}$$

$$T_I^i - T_I^{i+1} = 0, \quad \text{(C16)}$$

With these basic direct collocation constraints defined, we define a nonlinear program where we let $\mathbf{z} =$

[1] $y_1^d(\alpha)$ is typically a constant, often defining a steady forward velocity.

[2] These constraints encompass various practical constraints described in Section V-B, such enforcing no net moment about the foot, and ensuring non-negative ground reaction forces.

$\{z^i\}_{i\in\{0,1,2,\ldots,N\}}$ be a vector of all optimization variables, and state the optimization problem as,

$$\underset{\mathbf{z}^*}{\text{argmin}} \quad \mathcal{J}(\mathbf{z}) \tag{12}$$

$$\text{s.t} \quad \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}, \tag{13}$$

$$\mathbf{c}_{\min} \leq \mathbf{c}(\mathbf{z}) \leq \mathbf{c}_{\max}, \tag{14}$$

where $\mathcal{J}(\mathbf{z})$ is the scalar objective function, $\mathbf{c}(\mathbf{z})$ is a vector of functions defined in (C1)-(C16) organized in the order of nodes, $\mathbf{z}_{\min}$, $\mathbf{c}_{\min}$ and $\mathbf{z}_{\max}$, $\mathbf{c}_{\max}$ are the vectors containing the minimum and maximum values of optimization variables and constraints, respectively. We solve this NLP using IPOPT [35], which is well-suited for the large and highly sparse problems generated by this formulation. For DRC-HUBO, the NLP typically contains over 10,000 design variables and constraints. A more detailed accounting of all the above constraints can be found in [16].

While we have now effectively defined our desired outputs y_2^d, our actual outputs y_2^a can still take on any number of forms. Typically, these output forms have been linear in q to make online solving trivial. However, with modern inverse kinematics methods, we can potentially solve for very nonlinear outputs that are more task-relevant (e.g. foot position) at real-time speeds.

IV. Whole Body Kinematics

Whole body inverse kinematics is the process of finding at least one joint configuration that satisfies a set of simultaneous kinematic constraints for a body that has one or more limbs. This will often include multiple simultaneous end effector constraints, such as having two feet on the ground while grasping an object. In [30], this is approached as a control problem where each constraint is a task with a certain priority level. They use iterative methods which leverage the robot's Jacobian and the Jacobian's nullspace so that low priority constraints can be solved within the nullspace of higher priority constraints. The method solves for joint torques at each moment in time rather than explicitly computing reference joint positions. Conversely, in [21] they introduce a "floating base" configuration to the kinematics of the robot, allowing them to use iterative Jacobian methods to explicitly compute reference joint positions for all the joints in the robot. Since the DRC-HUBO is a stiff position-controlled robot, we prefer the latter approach for our application.

It is important to note that the iterative Jacobian methods have several weaknesses which make them unappealing. Particularly, they get stuck in local minima, are sensitive to solver parameters (*e.g.* step size and tolerance), and may require a large number of iterations before finding a solution. For an arbitrary humanoid kinematic model, the iterative Jacobian approach might be unavoidable. However, some humanoid robots, such as the DRC-HUBO, are specifically designed so that each limb offers a six degree-of-freedom closed-form solution for its inverse kinematics. The solutions for HUBO's limbs can be found in [1] and [24]. In our application, we can take advantage of these closed-form solutions to make the solving process fast and precise.

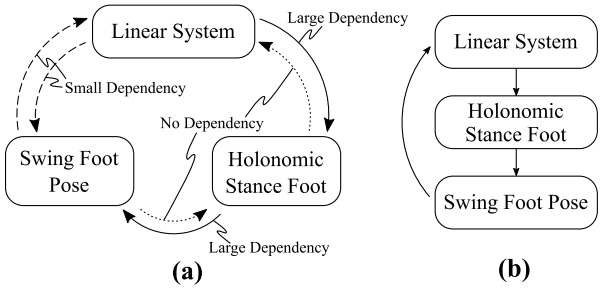

Fig. 2: **(a)** A dependency diagram of constraint types being solved by the whole-body inverse-kinematics solver. **(b)** The most efficient sequence for solving the constraints.

The kinematic constraints that we use take three forms. In the first constraint form, $\mathbf{e_d} - E\theta = 0$, θ is a vector of the joint positions, $\mathbf{e_d}$ is a vector function of time which is determined by the optimization routine discussed in Section III, and E is a coefficient matrix with fewer rows than columns. As a consequence, this is an underdetermined *linear system* of equations at each moment in time, and it can be solved for explicitly at each time step.

The second type of kinematic constraint is the *holonomic stance foot* constraint, $T_{stance}(\theta) = T_d$. which simply expresses that the homogeneous transform of the stance foot must remain at its desired location. We achieve this by computing the forward kinematics of the stance foot given its current configuration, then computing the "error transform" of its current configuration. We then adjust the floating base components of the robot's configuration to compensate for the error transform, thereby placing the stance foot where it belongs.

The final constraint is a *swing foot pose* constraint, $x_{dj} - x_j = 0$. In this case, x_j is a component of the swing foot's pose relative to the stance foot–represented as a translation vector and a set of Euler angles–and x_{dj} is the desired value for x_j. We formulate these constraints in terms of a standard Task Space Region [5], and we solve the Task Space Region constraint using the closed-form inverse kinematics solution for the six joints in the swing foot's leg. This allows us to solve the constraint directly and without any risk of getting caught in a local minimum.

The combination of the underdetermined linear system and swing-foot pose constraint fully determine the joint angles of the robot, while the stance-foot constraint fully determines the six degrees of freedom of the floating base. Even though the linear system of equations which can be solved explicitly, swing and stance constraints cannot be expressed as linear equations with respect to the joint positions, so the three constraint types cannot be merged into a single linear system of equations. However, all three constraint types have closed-form solutions which can each be solved explicitly. This allows us to solve each constraint type sequentially in order to find a solution that satisfies all three.

It is important to note that there is overlap between which

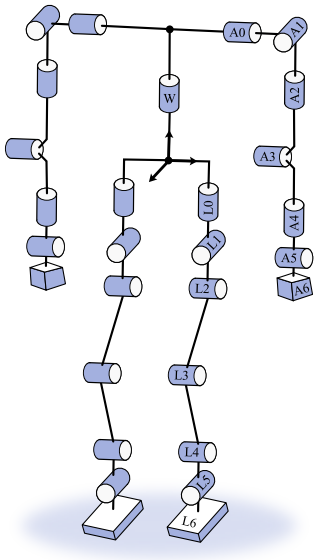

Fig. 3: Kinematic diagram of the DRC-HUBO platform.

components of the joint configuration are affected by each constraint type. This means that if you take the output of one constraint type's solution and pass it into a solver for the next constraint type, the new output might no longer satisfy the previous constraint. Therefore, when deciding on which order to use when solving the constraints, it is important to consider which components of the joint configuration will be affected by each constraint type. A diagram of these dependencies can be seen in Fig. 2a. The sequence that best takes advantage of the inter-dependencies between constraints is shown in Fig. 2b. Solving the constraints in this order usually allows the whole set of constraints to be satisfied within 2 - 7 cycles through the sequence. This allows us to generate trajectory waypoints at an average rate that exceeds 5 kHz, making this method suitable for real time control.

V. Application to DRC-HUBO

As a demonstration of our HZD+IK approach, we apply our method to the humanoid robotic platform, DRC-HUBO. We present simulations of two different stable full-body-optimized walking gaits (including arm motions) on a DRC-HUBO model in the DART simulation environment[3]. This includes an optimized startup motion to accelerate from rest to the steady periodic gait. We also present a preliminary implementation of one of these walking gaits on the DRC-HUBO robot using open-loop position control.

A. DRC-HUBO Model

The DRC-HUBO model used in this paper has 27 actuated degrees of freedom from its wrists to its feet [25][4], which are depicted in Fig. 3. We constrain our continuous domains to only allow flat-footed contact with the ground as to minimize

[3]The DART simulation environment is available at `https://github.com/dartsim/dart`.

[4]Fingers are excluded from dynamic optimization in this study.

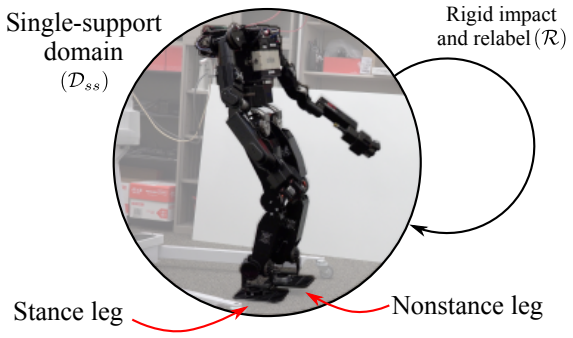

Fig. 4: Flow of domains for single-support walking with DRC-HUBO.

the necessary torques at the foot actuators. We thus model the domains of DRC-HUBO as a sequence of alternating single-support domains, $\mathcal{D}_{ss}$, as depicted in Fig. 4.

B. Optimization and Task Definitions

In addition to the base-level constraints defined in Section III, we also enforced a number of task-specific constraints on the PHZD optimization. For one, we ensure there is no net moment about the foot (note: we do not explicitly compute a zero moment point or enforce LIPM, see [13]). We impose impact velocity constraints of the nonstance foot, which limits potential damage to DRC-HUBO's drive mechanisms, as well as limits on joint rotation, velocity, and acceleration. Further, the feet must be separated by a minimum horizontal distance, and the nonstance foot height must exceed a clearance function above the ground.

We hypothesized that optimizing the robotic gait for energy economy would encourage DRC-HUBO to swing its arms. As such, we chose an objective to minimize the mechanical cost of transport over the course of the stride cycle:

$$\mathcal{J}(\mathbf{z}) := \frac{1}{mgd}\left(\sum_{i=0}^{N-1}\left(\frac{\|P(u^i,\dot{q}^i)\|\cdot\Delta t_i}{T_I^i}\right)\right), \qquad (15)$$

where mg is the robot weight and d is the total distance traversed by the center of mass. We assume no power regeneration, and thus the integrand $P(u^i,\dot{q}^i)$ is the total unsigned mechanical power consumed by the actuators. The optimization problem was formulated with 39 nodes[5].

In this study, we use our PHZD optimization to generate outputs for two types of motions: startup and periodic locomotion. We first optimize an efficient periodic gait subject to all listed constraints and objectives. Secondly, we optimize a startup motion, which requires all initial generalized coordinate velocities ($\dot{q}_0$) be set to zero, with final positions and velocities (q_f, $\dot{q}_f$) equal to our periodic gait.

We also tested this optimization method by generating two different periodic gaits. The first was allowed more liberal constraints on arm-joint velocities and effective foot size

[5]Note: this means there were effectively 20 central collocation points.

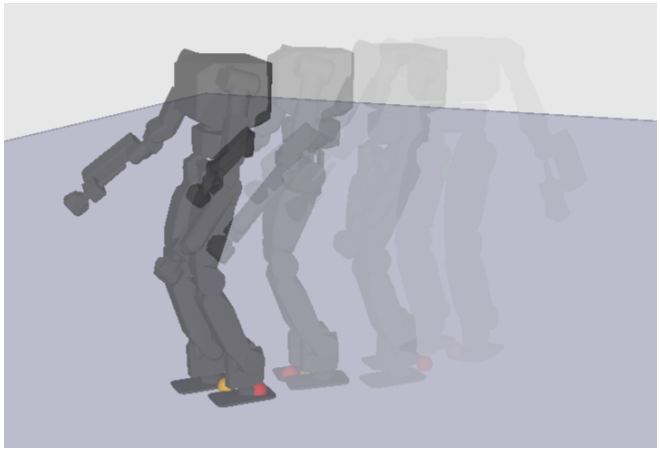

Fig. 5: A stroboscopic image of full-body-optimized stable walking in simulation (DART environment). The robot starts at rest, executes an optimized startup motion, and runs a pre-scripted open-loop trajectory representing optimized PHZD outputs for ten steps. Outputs are computed by inverse kinematics methods, which solve at an average rate of 5kHz.

as well as slower stepping frequencies[6]. The result of this optimization (**Gait A**) was a very natural-looking, counter-rotating arm swing, but was less likely to be successful on hardware. The second result (**Gait B**) was restricted by tighter constraints on arm-joint velocities, smaller effective foot size, and higher step frequencies. In addition to simulation, **Gait B** is presented with preliminary hardware results. Each of these gaits were solved with IPOPT (using the linear solver `ma57`) in approximately 7-10 minutes on a laptop computer (Intel Core i7-3820QM processor, 2.7 GHz, with 12 GB of RAM).

C. Simulation Results

We tested both **Gait A** and **Gait B** in the DART simulation environment. DART is a wholly different dynamics engine than the equations of motion seen by the optimizer to generate stable and dynamic gaits (for instance, DART uses LCP-based contact solving, while the optimizer sees an explicit formulation of the holonomic constraints). This makes for a good independent check that the optimizer is in fact generating stable walking controllers.

To test the stability of the generated controller, we enforce the designed outputs by solving the associated IK problem, and command the solved joint orientations via position control. As such, there is no feedback from the overall orientation of the robot, and the model must rely on the dynamic stability of the IK-solved open-loop trajectory[7]. After executing the optimized startup sequence, the robot was commanded to walk ten steps, playing back the trajectory generated by the inverse-kinematics solver.

Both **Gait A** and **Gait B** resulted in stable, dynamic walking in the DART simulator. Fig. 5 shows a stroboscopic image of **Gait B** walking in Cartesian space, while Fig. 7 shows a tiled comparison of the optimization dynamics and the DART simulation for both gaits. The exhibited gaits showed a bouncing center-of-mass behavior, not restricted to constant CoM heights as demanded by many LIPM-based methods, as plotted in Fig. 6 for **Gait B**. Further, the arms in **Gait A** exhibit a natural-looking counter-rotating swing.

[6]In open-loop position control, dynamic gaits tend to be more robust with higher stepping frequencies than typical humanoid gaits.

[7]The state-dependent phase variable τ is effectively replaced with time, so the resulting trajectory is fully time-dependent.

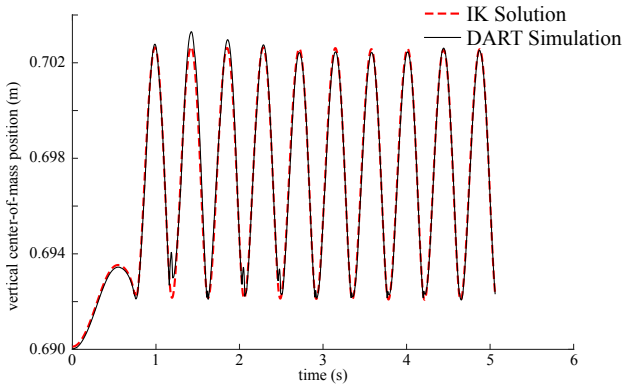

Fig. 6: Center-of-mass position over time for the simulated gait, compared to the raw kinematic trajectory computed via inverse kinematics (IK). This pendulum-like bouncing while walking demonstrates that the resulting gait is not restricted to a constant CoM height, as per LIPM-based techniques.

D. Preliminary Hardware Experiments

As a test of the framework's ability to produce stable control, we use only DRC-HUBO's encoders for position control feedback, and no inertial measurement. We fed the same open-loop IK-solved position trajectory from **Gait B** to DRC-HUBO. We noticed that the robot had a significant propensity to lean forward and fall during the beginning static position, which we determined was likely due to modeling errors. As such, we used the IK solver to adjust the gait intuitively, asking it to move the feet forward by $4cm$ to tune its balance. Because of the HZD-IK framework, this adjustment was computationally trivial. With this adjustment made, DRC-HUBO would start from rest, execute the startup procedure, and walk nine dynamic steps, open-loop, before falling forward. Fig. 8 shows tiled images of the experimental gait.

This preliminary experiment aims to show that these generated dynamic motions are reasonable to implement on a real humanoid robot, which are subject to physical limitations like motor saturation limits. Clearly, while stable in simulation, the result is not very robust to the modeling errors present. **Gait A**, which has more exaggerated arm swing, was also implemented on DRC-HUBO but had difficulty performing the startup maneuver without falling, despite its stability in the independent simulator. We suspect this failure is also likely due to inaccurate modeling of the link inertias, which are more drastically accelerated in **Gait A**. Future work will incorporate feedback control in order to compensate for modeling errors.

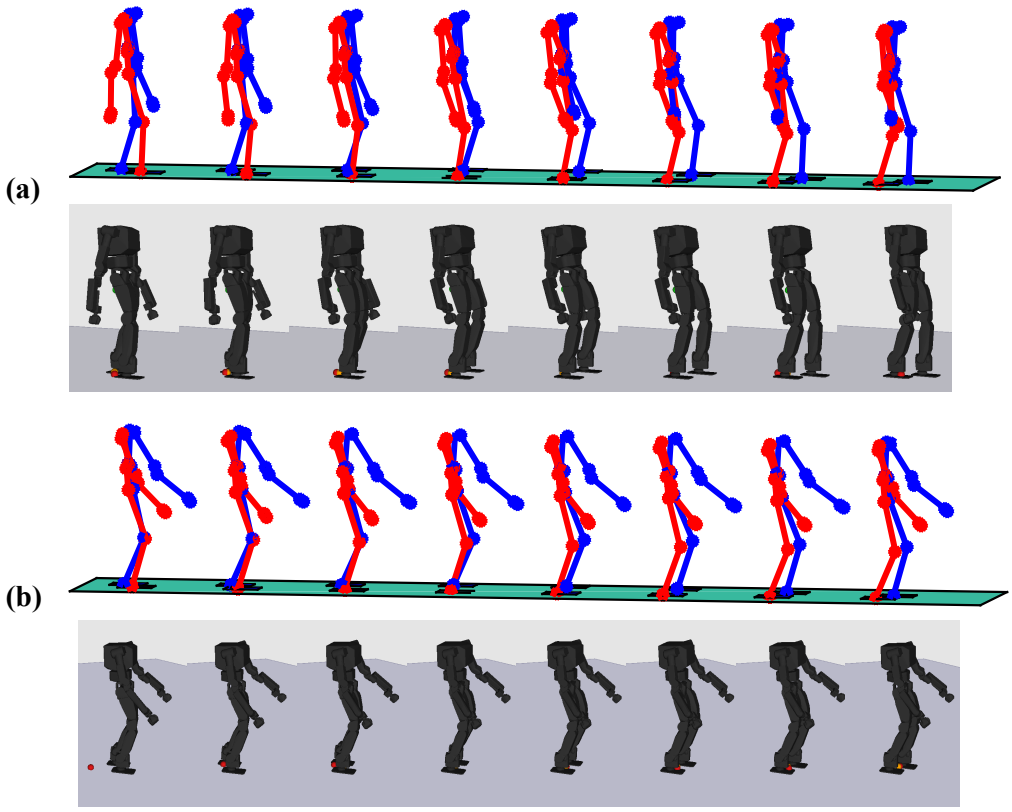

Fig. 7: A walking tiles figure comparing the optimized gait as simulated by the equations of motion used by the optimizer (top) to the open-loop walking controller simulated in DART (bottom) using optimized gaits after the startup procedure has finished: **(a) Gait A** and **(b) Gait B**. Both gaits were stable using open-loop position control and **Gait A** resulted in a solution with natural-looking counter-rotating arm swing.

VI. Conclusion

We presented simulation results with a model of the bipedal robot, DRC-HUBO, showing that hybrid zero dynamics approaches can scale up to the complexity of humanoid robots with swinging arms. With our offline large-scale (10,000-variable) optimization formulation, we were able to generate walking with dynamic arm swing in 7-10 minutes. One result demonstrated very natural human-like counter-rotation. Further, this arm swing was not defined *a priori* by a user, but emerged from a gait optimization considering a full multibody model of the dynamics. Specifically, the optimizer found this solution as a consequence of **1)** minimizing gait energy costs and **2)** maintaining no net moment about the foot, both of which are plausible from a biomechanical standpoint [7], [26].

Stable walking was achieved for two presented gaits using only open-loop trajectories obtained from the solver. These gaits were implemented online via a high-speed inverse kinematics solver, and verified in simulation through the independent simulator, DART. Future work will attempt to further accelerate the offline gait optimizations to be competitive with computation times from simplified LIPM-based gait planners. A preliminary implementation on DRC-HUBO, using just open-loop position trajectories, walked nine dynamic steps before falling, indicating a future need for feedback control.

Acknowledgment

The authors would like to thank their funding sources: DARPA grant number D15AP00006 and NSF grant numbers CPS-1239055 and NRI-1526519.

References

[1] Muhammad A. Ali, H. Andy Park, and C. S. George Lee. Closed-form inverse kinematics joint solution for humanoid robots. In *International Conference on Intelligent Robots and Systems*, pages 704–709, 2010.

[2] A. D. Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, May 2014.

[3] A. D. Ames, E. A. Cousineau, and M. J. Powell. Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control (HSCC)*, pages 135–144. ACM, 2012.

[4] Aaron D Ames. First steps toward automatically generating bipedal robotic walking from human data. In *Robot Motion and Control 2011*, pages 89–116. Springer, 2011.

[5] Dmitry Berenson, Siddhartha Srinivasa, and James Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *International Journal of Robotics Research*, 30(12):1435 – 1460, October 2011.

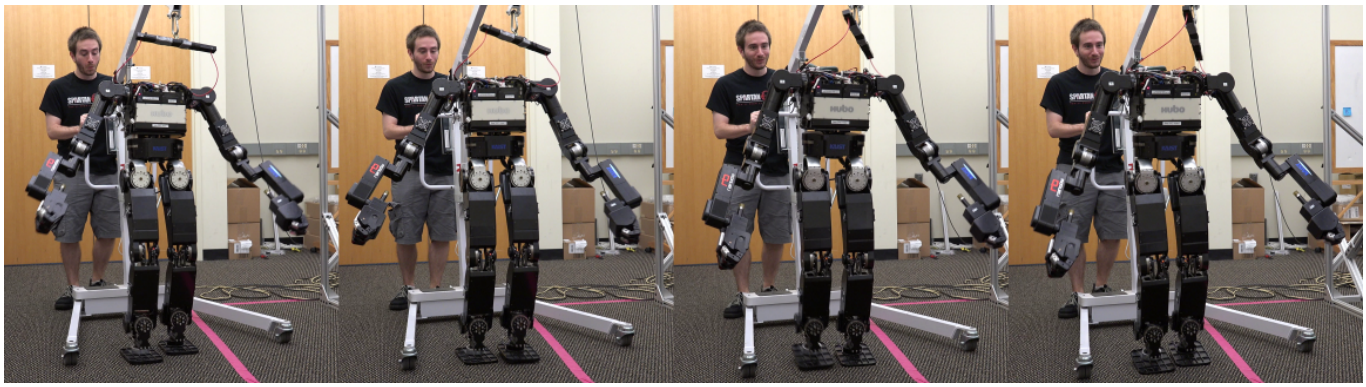

Fig. 8: Tiled images of a preliminary dynamic walking experiment with DRC-HUBO with an optimized open-loop controller. After a dynamic startup sequence, DRC-HUBO walked for nine steps before modeling errors caused a forward fall.

[6] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*, volume 19. SIAM, 2010.

[7] Steven H Collins, Peter G Adamczyk, and Arthur D Kuo. Dynamic arm swinging in human walking. *Proceedings. Biological sciences / The Royal Society*, 276(1673):3679–3688, 2009.

[8] Hongkai Dai, Andres Valenzuela, and Russ Tedrake. Whole-body Motion Planning with Simple Dynamics and Full Kinematics. In *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.

[9] Salman Faraji, Soha Pouya, Christopher G Atkeson, and Auke Jan Ijspeert. Versatile and Robust 3D Walking with a Simulated Humanoid Robot (Atlas): a Model Predictive Control Approach. In *IEEE International Conference on Robotics and Automation*, pages 1943–1950, 2014.

[10] Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization Based Full Body Control for the Atlas Robot. In *14th IEEE-RAS International Conference on Humanoid Robots*, 2014.

[11] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6):1–11, November 2013.

[12] J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. 46:51–64, January 2001.

[13] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955 – 1988, 2014.

[14] Charles R Hargraves and Stephen W Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.

[15] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online Walking Motion Generation with Automatic Footstep Placement. *Advanced Robotics*, 24(5-6):719–737, 2010.

[16] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *IEEE International Conference on Robotics and Automation (accepted)*, 2016.

[17] S Kajita, F Kanehiro, K Kaneko, K Fujiwara, K Harada, K Yokoi, H Hirukawa, and M Morisawa. Biped walking pattern generation by using preview control of zero-moment point. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1620–1626, Taipei, Taiwan, 2003.

[18] J Koenemann, A Del Prete, Y Tassa, E Todorov, O Stasse, M Bennewitz, and N Mansard. Whole-body Model-Predictive Control applied to the HRP-2 Humanoid. In *IEEE/RAS International Conference on Intelligent Robots and Systems*, 2015.

[19] Scott Kuindersma, Robin Deits, Maurice Fallon Andr, Hongkai Dai, Frank Permenter, Koolen Pat, and Marion Russ. Optimization-based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot. *Autonomous Robots*, 2015.

[20] Felipe Mathew DeDonato, Velin Dimitrov, Ruixiang Du, Ryan Giovacchini, Kevin Knoedler, Xianchao Long, Michael A. Gennert Felipe Polido, Taskin Padir, Siyuan Feng, Hirotaka Moriguchi, Eric Whitman, X. Xinjilefu, and Christopher G. Atkeson. Human-in-the-loop Control of a Humanoid Robot for Disaster Response: A Report from the DARPA Robotics Challenge Trials. *Journal of Field Robotics*, 32(2):275–292, 2015.

[21] Michael Mistry, Jun Nakanishi, Gordon Cheng, and Stefan Schaal. Inverse kinematics with floating base and constraints for full body humanoid robot control. In *International Conference on Humanoid Robots*, 2008.

[22] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-CIO: Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids. In *IEEE International Conference on Robotics and Automation*, 2015.

[23] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[24] Rowland O'Flaherty, Peter Vieira, M.X. Grey, Paul Oh, Aaron Bobick, Magnus Egerstedt, and Mike Stilman. Humanoid robot teleoperation for tasks with power tools. In *International Conference on Technologies for Practical Robot Applications*, 2013.

[25] Ill W. Park, Jung Y. Kim, Jungho Lee, and Jun H. Oh. Online free walking trajectory generation for biped humanoid robot KHR-3 (HUBO). volume 2006, pages 1231–1236, 2006.

[26] Jaeheung Park. Synthesis of natural arm swing motion in human bipedal walking. *Journal of Biomechanics*, 41(7):1417–1426, 2008.

[27] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus. Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid. *The International Journal of Robotics Research*, 31(10):1117–1133, August 2012.

[28] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[29] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer New York, 1999.

[30] Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *IEEE International Conference on Robotics and Automation*, 2006.

[31] Benjamin J. Stephens and Christopher G. Atkeson. Push Recovery by stepping for humanoid robots with force controlled joints. *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 52–59, December 2010.

[32] O. Von Stryk. *Numerical solution of optimal control problems by direct collocation*. Springer, 1993.

[33] M Vukobratovic and B Borovac. Zero-Moment Point–Thirty Five Years of its Life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[34] M Vukobratovic and D Juricic. Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transcations on*, (1):2–7, 1969.

[35] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[36] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press Boca Raton, 2007.