

Efficient HZD Gait Generation for Three-Dimensional Underactuated Humanoid Running

Wen-Loong Ma¹, Ayonga Hereid¹, Christian M. Hubicki¹ and Aaron D. Ames²

Abstract—Dynamic humanoid locomotion is a challenging controls problem, and running is especially difficult to achieve, given the underactuation inherent to aerial domains. Previous work developed a gait-generating optimization framework for dynamic locomotion in the context of hybrid zero dynamics, producing stable 3D walking on the humanoid hardware platform, DURUS. Here, we demonstrate that this optimization method also extends to stable 3D running, despite the exacerbated underactuation. Gaits generated from the optimization, which optimizes the dynamics of all 23 degrees of freedom to maximize energy economy, result in stable running in a DURUS simulation model. Notably, the presented running is underactuated in all domains, due to DURUS’ spring-legged design. Further, we generated 25 different running gaits, over a range of speeds (1.5-3.0 m/s), to demonstrate the reliability of solving the large-scale nonlinear program. We report statistical performance of the optimization in successfully generating stable running (average computation time: 323 seconds) in an effort to establish a benchmark for large-scale gait generation. We inspected this array of gaits across speeds, noting recognizable trends in optimized strategies from prior studies on lower-order models, (e.g. both increased step frequency and step length with speed), along with the first reported cost-of-transport curve for a 3D humanoid running model. We consider this result an important step toward humanoid running on the DURUS hardware platform.

I. INTRODUCTION

In 1989, in Marc Raibert’s technical report “Dynamically Stable Legged Locomotion” [22], it was summarized that “*The running speed of a legged system depends upon the frequency and length of its steps*” based on the study of the legendary hopper. To better understand this phenomenon in 3D bipedal legged systems, this paper shows a statistical result based on all the stable running gaits obtained from the optimization based gait generation framework, i.e. how should the optimizer respond when asked to generate faster gait with the goal of minimizing the cost of transport. Bipedal running is an important benchmark for humanoid control for a number of mathematical and practical reasons. Unlike walking, running is an inherently underactuated control problem [27]. Whenever the robot leaves the ground, it fundamentally loses its ability actuate all of its degrees of freedom, and is at the mercy of its ballistic trajectory. It is also a multi-domain hybrid control problem [36], since both

*This research is supported by CPS grant 1239085 and SRI grant W31P4Q-13-C-009.

¹W. Ma, A. Hereid, and C. M. Hubicki are with the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, {wenlongma, ayonga27, chubicki6}@gatech.edu.

²A. D. Ames is with the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, ames@gatech.edu.



Fig. 1: The humanoid robot, DURUS, running stably in 3D in a simulation environment as a result of a large-scale hybrid zero dynamics optimization. This study presents the result of 25 gait optimizations achieving a range of running speeds.

walking and running should be modeled with continuous dynamics. Further, the higher forces and power demands push the practical limits of humanoid actuators. Here, we present 3D running via hybrid zero dynamics (HZD) [33] on a simulated underactuated model of the humanoid robot, DURUS [24]. The running gaits emerge directly from a large-scale gait optimization of the full-order system dynamics, a tool which was previously developed for 3D walking with the DURUS hardware [13]. We report the success of this toolset as a milestone toward running control on the DURUS humanoid.

The earliest example of running controllers were developed using set of highly successful heuristics. Examples include the the Raibert hoppers [21] and the ARL-Monopod II [1]. Decades later, Honda’s humanoid robot, ASIMO [26], claims running speeds up to 2.5 m/s although its control methods are not reported. Other methods have been employed which achieve stable running in simulation by constraining the robot’s dynamics to a reduced-order model [16] and even achieving simulated high-speed turning [32]. We seek a method which generates stable gaits as a result of optimizing the full dynamical equations of the multibody humanoid.

Researchers have also generated simulated running gaits for simulated robots with various degrees of freedom, from simple point-mass models [30] to planar hopping models [34]

is defined as a *tuple* [2]:

$$\mathcal{H}\mathcal{C} = (\Gamma, \mathcal{D}, \mathcal{U}, S, \Delta, FG), \quad (1)$$

where

- $\Gamma = \{V, E\}$ is a directed cycle with vertices $V = \{s, f\}$, where s represents the stance domain and f represents the flight domain, and the edges $E = \{s \rightarrow f, f \rightarrow s\}$,
- $\mathcal{D} = \{\mathcal{D}_s, \mathcal{D}_f\}$ is a set of admissible domains of continuous dynamics,
- $\mathcal{U} = \{\mathcal{U}_s, \mathcal{U}_f\}$ is a set of admissible controls,
- $S = \{S_s \subset \mathcal{D}_s, S_f \subset \mathcal{D}_f\}$ is a set of *guards*,
- $\Delta = \{\Delta_{s \rightarrow f}, \Delta_{f \rightarrow s}\}$ is a set of smooth *reset maps* represents the discrete dynamics,
- $FG = \{(f_s, g_s), (f_f, g_f)\}$ is a set of affine control systems, $\dot{x} = f_v(x) + g_v(x)u$, defined on \mathcal{D}_v for all $v \in V$, with $x = (q_e, \dot{q}_e)$ be the system states.

The directed cycle Γ is depicted in the Fig. 3. The construction of individual elements of (1) will be presented in the remainder of this section.

Stance Domain. During the stance domain, the stance foot remains flat on the ground. Often we use holonomic constraints to model the foot contact with the ground [9]. In particular, we define the holonomic constraints of the stance domain as

$$\eta_s(q_e) := (p_{sf}(q_e), \phi_{sf}(q_e)) \in \mathbb{R}^6 \quad (2)$$

where p_{sf} the Cartesian position and ϕ_{sf} the orientation of the stance foot. Given the mass, inertia, length and the center of mass position of each link, the unconstrained dynamical equation of the stance domain \mathcal{D}_s is given by

$$D(q_e)\ddot{q}_e + H(q_e, \dot{q}_e) = Bu + J_s^T(q_e)F_e, \quad (3)$$

where, $D(q_e)$ is the inertia matrix, $H(q_e, \dot{q}_e)$ contains the Coriolis, gravity and spring forces terms, $J_s(q_e)$ is the Jacobian of the holonomic constraints, and F_e is a *wrench* containing the ground constraint forces and moments. The holonomic constraints are guaranteed via enforcing the second order derivative of η_s to be zero:

$$J_s(q_e)\ddot{q}_e + \dot{J}_s(q_e, \dot{q}_e)\dot{q}_e = 0, \quad (4)$$

Thus the affine control system $\{f_s, g_s\}$ can be determined by combining (3) and (4).

The manifold of the stance domain is determined by unilateral constraints, which could be formulated as a vector of admissible conditions, $A_s(q_e, \dot{q}_e, u)$. These conditions includes positive non-stance foot height, positive normal ground force, etc. In other words, we have

$$\mathcal{D}_s = \{(q_e, \dot{q}_e, u) \in T\mathcal{Q}_e \times U_s : A_s(q_e, \dot{q}_e, u) \geq 0\}. \quad (5)$$

Further the guard condition of the stance domain is defined as the normal ground force crosses zero, i.e.,

$$S_{s \rightarrow f} = \{(q_e, \dot{q}_e, u) \in T\mathcal{Q}_e \times U_s : F_e^z(q_e, \dot{q}_e, u) = 0\}. \quad (6)$$

The reset map from the stance domain to flight domain is an identity map, i.e., $\Delta_{s \rightarrow f} = \mathbf{I}$, considering the fact that there is no impact involved during the transition.

Flight Domain. There is no ground contact during the flight domain, therefore, the continuous dynamics of the domain is determined simply by the unconstrained Euler-Lagrangian equation:

$$D(q_e)\ddot{q}_e + H(q_e, \dot{q}_e) = Bu. \quad (7)$$

The admissible conditions of the flight domain are defined so that both feet are above the ground, i.e., $A_f(q_e) = (h_{sf}(q_e), h_{nsf}(q_e))$. Therefore, we have

$$\mathcal{D}_f = \{(q_e, \dot{q}_e, u) \in T\mathcal{Q}_e \times U_f : A_f(q_e) \geq 0\}. \quad (8)$$

Accordingly, the transition from the flight to stance domain occurs when the non-stance foot strikes the ground, i.e.,

$$S_{f \rightarrow s} = \{(q_e, \dot{q}_e, u) \in T\mathcal{Q}_e \times U_f : h_{nsf}(q_e) = 0, \dot{h}_{nsf}(q_e) < 0\}. \quad (9)$$

The reset map from the flight to stance domain incorporates the impact dynamics when the non-stance foot hits the ground, during which the joint velocities undergo discrete changes due to the introduction of new contact constraints. Given the pre-impact states (q_e^-, \dot{q}_e^-) , the post impact states $(q_e^+, \dot{q}_e^+) = \Delta_{f \rightarrow s}(q_e^-, \dot{q}_e^-)$ are determined by assuming a perfectly plastic impact of the rigid body model [9], [14]. Let \mathcal{R} be the coordinates relabeling matrix due to the change of the stance and non-stance legs, we have $q_e^+ = \mathcal{R}q_e^-$, and the plastic impact equation which determines the discrete changes of velocities, given as

$$\begin{bmatrix} D(q_e^-) & -J_s^T(q_e^-) \\ J_s(q_e^-) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_e^+ \\ \delta F_e \end{bmatrix} = \begin{bmatrix} D(q_e^-)\dot{q}_e^- \\ 0 \end{bmatrix}, \quad (10)$$

where δF_e is a vector of impulsive contact wrenches.

C. Hybrid Zero Dynamics (HZD) Control Framework

Given the hybrid control system model, we now present the *hybrid zero dynamics* framework in which virtual constraints are employed as a method to synthesis feedback controller to render stable running behavior of DURUS.

Virtual Constraints. Any admissible state-based feedback controller that has been applied to the control system, FG , yield closed-loop hybrid system [2]. This can be done by defining a set of virtual constraints—also referred as *outputs*, which is the difference between actual and desired outputs—and applying feedback controllers to drive them to zero [33].

Actual outputs $y^a(q_e, \dot{q}_e)$ are defined as functions of system states. For the stance domain \mathcal{D}_s , the forward velocity of the center of mass (CoM) is chosen as the relative degree one output $y_{1,s}^a = v_{com}^x(q_e, \dot{q}_e)$ to regulate the forward velocity of the robot, and the (vector) relative degree two outputs are defined as $y_{2,s}^a(q_e) = (\theta_{sk}, \phi_{slr}, \theta_{sh}, \psi_{sh}, \phi_{sa}, \phi_w, \theta_w, \psi_w, \theta_{nsk}, \phi_{nsh}, \theta_{nsh}, \phi_{nsf}, \theta_{nsf}, \psi_{nsf})$. For the flight domain \mathcal{D}_f , actual outputs consist of only the relative degree two outputs, given as $y_{2,f}^a(q_e) = (\theta_{sk}, \phi_{slr}, \theta_{sh}, \phi_{sf}, \theta_{sf}, \psi_{sf}, \phi_w, \theta_w, \psi_w, \theta_{nsk}, \phi_{nsh}, \theta_{nsh}, \phi_{nsf}, \theta_{nsf}, \psi_{nsf})$. In particular, $\phi_{slr} = \phi_{sh} - \phi_{nsh}$ is the stance leg roll, $(\phi_{sf}, \theta_{sf}, \psi_{sf})$ and $(\phi_{nsf}, \theta_{nsf}, \psi_{nsf})$ are the 3-dimensional orientations (Euler angles) of the stance and

non-stance foot link, respectively. Other outputs are joint angles as shown in Fig. 2.

We define that the desired velocity of CoM is parameterized by a constant v_d , i.e., $y_{1,s}(q_e, \dot{q}_e, v_d) = y_{1,s}^a(q_e, \dot{q}_e) - v_d$. Desired relative degree two outputs $y_{2,v}^d(\tau(q_e), \alpha_v)$ represented by 7th order Bézier polynomials [33] parameterized by a set of parameters α_v with $v \in \{s, f\}$. The virtual constraints on \mathcal{D}_v then can be defined as:

$$y_{2,v}(q_e, \alpha_v) = y_{2,v}^a(q_e) - y_{2,v}^d(\tau(q_e), \alpha_v), \quad (11)$$

where $\tau(q_e)$ is a monotonic state-based parameterization of time, defined as $\tau(q_e) = \frac{p_b^x - (p_b^x)^+}{(p_b^x)^- - (p_b^x)^+} \in [0, 1]$, where p_b^x is the x -position of the frame R_b , and $(p_b^x)^+$ and $(p_b^x)^-$ are the value of p_b^x at the beginning and the end of one step. In particular, the desired outputs of the stance and non-stance foot orientations are set to be zero respectively to keep the feet being flat throughout the step. If a output is defined on both stance and flight domains, the coefficients of the corresponding desired Bézier polynomials must be the same.

Feedback Controller. To drive the virtual constraints $y_v = (y_{1,v}, y_{2,v}) \rightarrow 0$ exponentially for each $v \in \{s, f\}$, we utilize the feedback linearization control law

$$w_v^\varepsilon = -\mathcal{A}_v^{-1} ((\mathcal{L}_f^2)_v + \mu_v^\varepsilon) \quad (12)$$

where \mathcal{A}_v is the decoupling matrix, given by

$$\begin{aligned} \mathcal{A}_s &= [\mathcal{L}_{g_s} y_{1,s}(q_e, \dot{q}_e); \mathcal{L}_{g_s} \mathcal{L}_{f_s} y_{2,s}(q_e)], \\ \mathcal{A}_f &= [\mathcal{L}_{g_f} \mathcal{L}_{f_f} y_{2,f}(q_e)] \end{aligned}$$

respectively, and

$$(\mathcal{L}_f^2)_s = [0; \mathcal{L}_{f_s} \mathcal{L}_{f_s} y_{2,s}(q_e)], \quad (\mathcal{L}_f^2)_f = [\mathcal{L}_{f_f} \mathcal{L}_{f_f} y_{2,f}(q_e)],$$

with \mathcal{L} is the Lie derivative. With the given control law, we have the output dynamics $(\dot{y}_{1,s}, \ddot{y}_{2,s}) = -\mu_s^\varepsilon$ and $\ddot{y}_{2,f} = -\mu_f^\varepsilon$ for the stance and flight domain respectively, where μ_v^ε can be chosen so that the outputs converge to zero exponentially at a rate of $\varepsilon > 0$. In particular, we define

$$\mu_s^\varepsilon = \begin{bmatrix} \varepsilon y_{1,s}(q_e, \dot{q}_e, v_d) \\ 2\varepsilon \dot{y}_{2,s}(q_e, \dot{q}_e, \alpha_s) + \varepsilon^2 y_{2,s}(q_e, \alpha_s) \end{bmatrix}, \quad (13)$$

$$\mu_f^\varepsilon = 2\varepsilon \dot{y}_{2,f}(q_e, \dot{q}_e, \alpha_f) + \varepsilon^2 y_{2,f}(q_e, \alpha_f). \quad (14)$$

Partial Hybrid Zero Dynamics. Moreover, the control law in (12) renders the *full zero dynamics surface* exponentially stable and invariant over both continuous domain [2]. Due to the impact dynamics of the non-stance foot, the invariance of the full zero dynamics surface is no longer guaranteed. Particularly, it would be infeasible to enforce the relative degree one output to be invariant through impact due to the changes in the velocity. It motivates us to consider the *partial hybrid zero dynamics* for the running of DURUS specifically. Therefore, the goal of designing a periodic and dynamic running gait is to find a set of parameters $\alpha = \{v_d, \alpha_s, \alpha_f\}$ that ensures there exists a periodic orbit for the system in (1) and the partial zero dynamics surface,

$$\mathcal{PZ}_v = \{(q_e, \dot{q}_e) \in \mathcal{D}_v : y_{2,v}(q_e) = 0, \dot{y}_{2,v}(q_e, \dot{q}_e) = 0\},$$

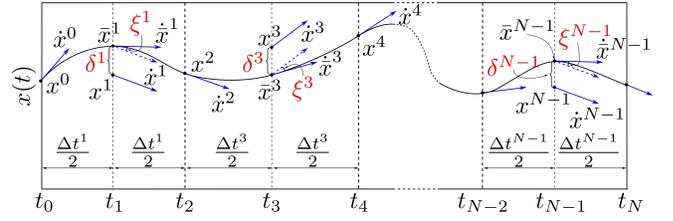


Fig. 4: Illustration of defect constraints and node distribution of the direct collocation method.

is invariant through impact, where $v \in \{s, f\}$. The process of finding α is often formulated as a nonlinear constrained optimization problem subject to the multi-domain hybrid system model and HZD control framework of bipedal running.

III. RUNNING GAIT OPTIMIZATION

This section will emphasize the optimization method we utilized to design periodic running gaits for the 3D humanoid robot DURUS. One popular approach to generate locomotion gaits that satisfy hybrid zero dynamics conditions, direct single shooting method [23] based nonlinear programming (NLP), has been widely utilized on the subjects including 2D walking [2], [28], multi-domain walking [36] and 2D running [37], [29]. In special cases such as fully actuated robotic walking[20], and planar point feet robot [35], the closed form solution of zero dynamics can be computed analytically. This makes the single shooting problem trivial to solve in the optimization algorithm.

However, for bipedal running, due to the multiple degrees of underactuation involved in the multi-domain hybrid system, it is computational expensive and time consuming to forward integrate on the zero dynamics surface to find the time solution to use the direct method inside the NLP. Thus a direct collocation method based NLP algorithm is used to solve a periodic running gait satisfies the equality and inequality constraints. This methodology has been applied successfully on the DURUS 3D walking biped [13] and this section will present this framework with an emphasis on running.

A. Formulation of Direct Collocation Optimization

Direct collocation method has been widely employed by trajectory optimization problems due to its effectiveness and robustness. Instead of explicitly integrating the dynamical system, the direct collocation method uses implicit Runge-Kutta methods to approximately determine the solution of the system. In the direct collocation method, the solution of a continuous domain, $\mathcal{D}_v \in \{\mathcal{D}_s, \mathcal{D}_f\}$, is discretized based on the time discretization

$$0 = t_0 < t_1 < t_2 < \dots < t_{N_v} = T_{I,v}, \quad (15)$$

assuming $T_{I,v} > 0$ is the time at which the system reaches the guard associated with a given domain. In this paper, we utilize the Hermite-Simpson (Separated) scheme (see [3]) with a novel modification via introducing defect variables. As illustrated in Fig. 4, we call the even nodes as *cardinal*

nodes, and the odd nodes as *interior nodes*. It is required that the number of cardinal nodes, N_v^c , of one domain has to be greater than 2. For each continuous domain, the number of segments is determined by $N^v = 2(N_v^c - 1)$. In particular, we place cardinal nodes on Chebyshev-Gauss-Lobatto (CGL) points, and place interior points at the middle point of two adjacent cardinal nodes. The solutions between two neighboring cardinal nodes are approximated by cubic interpolation polynomials [12], whose coefficients could be determined by the discrete states and their derivatives of the associated cardinal nodes. Then the NLP can be formulated so as to find a set of discrete system states that satisfies defect constraints of the implicit integration scheme.

Defect Constraints. Before defining defect constraints for the direct collocation scheme, we introduce another important ingredient of the optimization problem formulation—*defect variables*. Defect variables are supplementary decision variables that could be computed in closed-form initially, e.g., \dot{x} via *FG*. The purpose of introducing defect variables is to simplify the constraint expression, so that determining the analytical first-order Jacobian of constraints becomes feasible [13]. In large-scale sparse NLPs, providing analytical Jacobian of constraints would significantly increase the computation speed and improve the robustness the optimization convergence. Given the definition of decision variables, two defect constraints are defined at each interior node $i \in \{1, 3, 5, \dots, N_v - 1\}$ for all $v \in \{s, f\}$:

$$\xi^i = \dot{x}^i - \dot{\hat{x}}^i = 0 \quad (16)$$

$$\delta^i = x^i - \hat{x}^i = 0 \quad (17)$$

where $x^i = (q_e^i, \dot{q}_e^i)$ and $\hat{x}^i = (\hat{q}_e^i, \hat{\dot{q}}_e^i)$, and $\dot{\hat{x}}^i = 3(x^{i+1} - x^{i-1})/2\Delta t_v^i - (\dot{x}^{i-1} + \dot{x}^{i+1})/4$ and $\hat{x}^i = (x^{i+1} + x^{i-1})/2 + \Delta t_v^i(\dot{x}^{i-1} - \dot{x}^{i+1})/8$ respectively, with $\Delta t_v^i = t_{i+1} - t_{i-1}$ being the time interval [12].

Dynamics Constraints. Here, \hat{x}^i , more precisely \hat{q}_e^i , is introduced as decision variables of the NLP, which are determined by the continuous dynamics equations defined in (3), (4) and (7). In general, we enforce the following equality constraints at each node i of domain $v \in \{s, f\}$:

$$D(q_e^i)\hat{q}_e^i + H(q_e^i, \dot{q}_e^i) - Bu^i - J_v^T(q_e^i)F_e^i = 0, \quad (18)$$

$$J_v(q_e^i)\hat{q}_e^i + \dot{J}_v(q_e^i, \dot{q}_e^i)\hat{q}_e^i = 0. \quad (19)$$

It can be noted that the constraint in (19) is only enforced for the stance domain, and (implicitly) determines the wrenches F_e^i of the ground contact. The main advantage of this formulation over the traditional approach—computing \dot{x}^i via *FG*—is that it uses simpler but equivalent equality constraints in the optimization. Because deriving the closed form expression of *FG* requires the inversion of inertia matrix $D(q_e)$, which makes symbolically generating the analytical Jacobian of the defect constraints extremely unlikely for high DoF robots.

Moreover, system states (q_e, \dot{q}_e) at the first and last node should be consistent through the corresponding reset maps, so that the resulting gait is periodic. The consistency constraints from the stance domain to flight domain are trivial due to $\Delta_{s \rightarrow f}$ being an identity matrix, whereas the constraints

from the flight domain to stance domain must satisfy the coordinates relabeling and the impact equation in (10) that define the reset map $\Delta_{f \rightarrow s}$.

Manifold Constraints. As defined in Sect. II-A, system states and control inputs of each continuous domain must be in the domain manifold, \mathcal{D} . In other words, they must satisfy the admissible constraints of corresponding domain given in (5) and (8). For the stance domain, the admissible constraints also depend on ground contact wrenches. In our defect variables formulation, we could impose constraints on them directly due to the fact that these wrenches are introduced as decision variables.

Further, guard conditions defined in (6) and (9) must be enforced at the last node of corresponding domain to guarantee that the resulting gait indeed hits the guard of each domain at its last node.

HZD Constraints. It can be noted that the feedback controllers u^i in (18) are also introduced as decision variables and determined by the desired output dynamics given in (13) and (14), i.e.,

$$\begin{bmatrix} \dot{y}_{1,s}(q_e^i, \dot{q}_e^i, \ddot{q}_e^i, v_d) \\ \ddot{y}_{2,s}(q_e^i, \dot{q}_e^i, \ddot{q}_e^i, \alpha_s^i) \end{bmatrix} - \mu_s^\varepsilon(q_e^i, \dot{q}_e^i, v_d^i, \alpha_s^i) = 0 \quad (20)$$

$$\ddot{y}_{2,f}(q_e^i, \dot{q}_e^i, \ddot{q}_e^i, \alpha_f^i) - \mu_f^\varepsilon(q_e^i, \dot{q}_e^i, \alpha_f^i) = 0 \quad (21)$$

for all $i \in \{0, 1, 2, \dots, N_v\}$ with $v \in \{s, f\}$. Unlike traditional trajectory optimization problems, in which the control inputs often use open loop control policy, we incorporate the HZD based feedback control framework in our running gait optimization. Further, we enforce the desired output dynamics instead of using the closed form expression of u^i as in (12), resulting in much simpler expression for the constraints.

More importantly, the hybrid invariance condition of the resulting periodic gait becomes straightforward with this formulation. Specifically, the relative degree two outputs $y_{2,v}$ and their derivatives $\dot{y}_{2,v}$ should be enforced zero at the beginning of each domain. In other words, we define

$$y_{2,v}(q_e^0, \alpha_v^0) = 0, \quad (22)$$

$$\dot{y}_{2,v}(q_e^0, \dot{q}_e^0, \alpha_v^0) = 0. \quad (23)$$

at the first node of each domain $v \in \{s, f\}$. Additionally, gait parameters $T_{I,v}$ and α are defined at all nodes albeit being constants. Therefore, equality constraints between two neighboring nodes must be enforced to ensure the constancy of parameters. A constancy constraint is also imposed between the stance and flight domains for any outputs that defined on both domains. Incorporating all constraints discussed above along with additional physical constraints, such as joint angle and velocity limits and torque limits, the goal of generating energy efficient HZD running gaits for the 3D underactuated humanoid can be stated as a large-scale sparse NLP [13].

B. Energy Efficient Bipedal Running Optimization

We start by defining the decision variables for the problem. Let $\mathbf{Z}_v = (z_v^0, z_v^1, \dots, z_v^{N_v})$ be a vector of decision variables for $v \in \{s, f\}$ with z_v^i given as $z_v^i =$

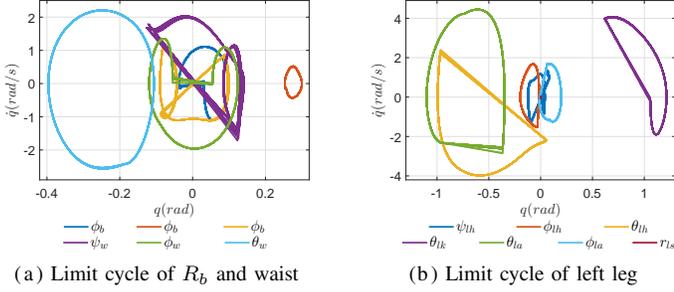


Fig. 5: Limit cycle of running at 2.0 m/s over 20 steps.

$(T_{I,v}^i, q_e^i, \dot{q}_e^i, \ddot{q}_e^i, u^i, F_e^i, \alpha_v^i, v_d^i)^1$, then $\mathbf{Z} = \{\mathbf{Z}_s, \mathbf{Z}_f\}$ is the vector of all decision variables of the NLP. Hence, the HZD based energy efficient running gait optimization problem for DURUS is stated as,

$$\mathbf{Z}^* = \underset{\mathbf{Z}}{\operatorname{argmin}} \sum_{v=\{s,f\}} \mathcal{J}_v(\mathbf{Z}_v) \quad (24)$$

$$\text{s.t.} \quad \mathbf{Z}_{\min} \leq \mathbf{Z} \leq \mathbf{Z}_{\max}, \quad (25)$$

$$\mathbf{C}_{\min} \leq \mathbf{C}(\mathbf{Z}) \leq \mathbf{C}_{\max}, \quad (26)$$

where $\mathcal{J}_v(\mathbf{Z}_v)$ is the cost function which minimizes the specific cost of transport of the running gait, given by

$$\mathcal{J}_v(\mathbf{Z}_v) := \frac{1}{mgd} \left(\sum_{i=0}^{N_v-1} \left(\frac{\|P_v(u^i, \dot{q}_e^i)\| \cdot \Delta t_i}{T_{I,v}^i} \right) \right), \quad (27)$$

where $\Delta t_i = t_{i+1} - t_i$, mg is the robot weight, d is the distance traveled during a gait cycle, and $P_v(u^i, \dot{q}_e^i)$ is the total power consumed (assuming no power-regeneration) computed at each segment. All constraints defined in Sect. III-A are formulated in a vector form $\mathbf{C}(\mathbf{Z})$, with \mathbf{C}_{\min} and \mathbf{C}_{\max} are vectors of the minimum and maximum values of constraints, respectively. For equality constraints, the corresponding minimum and maximum values are set to zero. Moreover, \mathbf{Z}_{\min} and \mathbf{Z}_{\max} are the vectors of the minimum and maximum values of all decision variables. Hence, some physical constraints, such as joint angle and velocity boundaries as well as the maximum torque limits of the robot hardware can be imposed directly as the boundary values of the decision variables.

The result is a large-scale NLP problem that generates energy efficient periodic running gaits for DURUS. With the defect variables formulation, the analytical Jacobian of the constraints can be computed by proper symbolic mathematics toolboxes. Often, the Jacobian matrix of the whole NLP constraints is very sparse—the density is far less than 1%—which allows the problem to be solved efficiently using appropriate large sparse NLP solvers such as IPOPT [31].

IV. OPTIMIZATION & SIMULATION RESULTS

By using the proposed optimization method, we are able to generate multiple stable 3D running gaits for DURUS with forward velocity varying from 1.5 m/s to 3.0 m/s. This

¹ F_e^i and v_d are only defined at nodes of the stance domain by definition.

section will emphasize one of the simulated running gaits in detail first, and then a performance report of the optimization algorithm and an overall statistical analysis of all the running gaits we have generated.

A. Running at 2.0 m/s

With the constraints configured as explained in Sect. III and using the large-scale IPOPT NLP solver developed by “COIN-OR”, a 3D running gait was generated after 722 iterations and 374 seconds of computation, with dual infeasibility converging to $9.0e^{-4}$, and constraints violation $1.8e^{-7}$. This particular gait runs is at 2.0 m/s. Note that we categorize each running gait based on the COM velocity along the x direction during the flight domain since it is a constant. The specific cost of transport (SCOT) [5] is computed in simulation as 0.90, the maximum angular velocity of all joints is 4.4 rad/s, peak torque is 446 N m and peak power is 1.1 kW. A running tile is also shown at Fig. 8, the limit cycle of each joint is also shown as Fig. 5. Only one leg is shown because of the symmetric motion. Moreover, as we have verified the stability of the running gait by numerically computing the eigenvalues of the linearization of the Poincaré map restricted to the zero dynamics about the Poincaré section where $p_b^x = 0$, the magnitude of its eigenvalues are shown to be $[0.4138, 0.0827, 0.0312, 0.0062, 0.0000]$. With all values smaller than 1, this indicates asymptotic stability can be obtained from this running framework (see [18] for more detail).

B. Efficient 3D Running Gaits Generation

The main contribution of this paper is to present a working framework to generate stable running gaits for 3D bipedal robots not only reliably, but also efficiently. To illustrate the computational efficiency and reliability of this framework, the computation time and iteration the optimization required are shown on Fig. 7 for the 25 gaits the optimization has found. As a result, it takes 609 iterations and 323 seconds on average for the optimization to find a feasible solution². Note that the threshold of the dual infeasibility of the NLP is set to be $1e^{-3}$, the number of grid is chosen to be 15 for both the stance and flight domains. All constraints and physical limitations are configured based on the hardware capability and the constraint violation converged below $1e^{-6}$.

V. DISCUSSION

Due to the high degrees of underactuation and the change in dimension of the hybrid zero dynamics (2 for stance domain and 5 for flight domain) through a step, direct integration over time to find the solution can raise the computation complexities dramatically, which can make the optimization infeasible. However, despite the high DOF of 3D running dynamics, by simply modifying the forward velocity constraint and fine tuning few constraint boundaries to adjust the running appearance, this gait generation method

²This algorithm is running on a Ubuntu14.04.4 LTS desktop, equipped with an Intel® Xeon® processor E3-1246 V3 (3.5GHz).

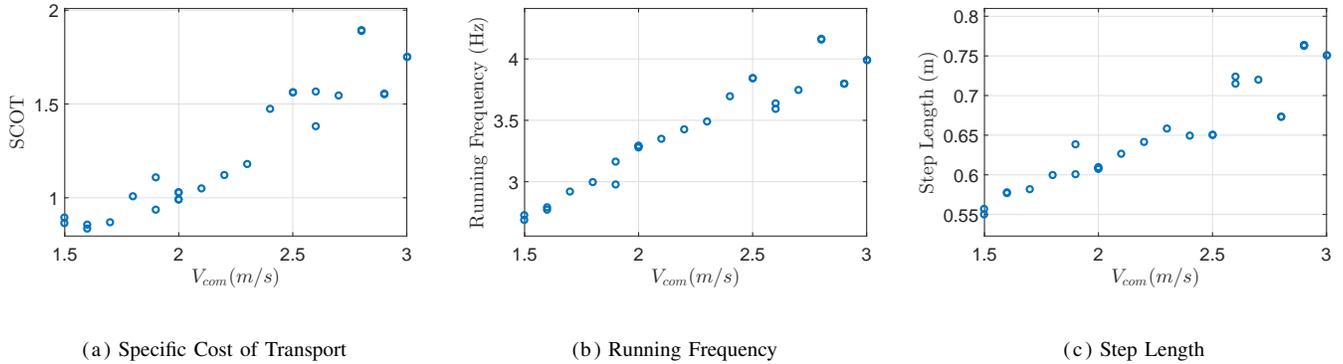


Fig. 6: Multiple running gaits with forward velocity from 1.5 m/s to 3.0 m/s.

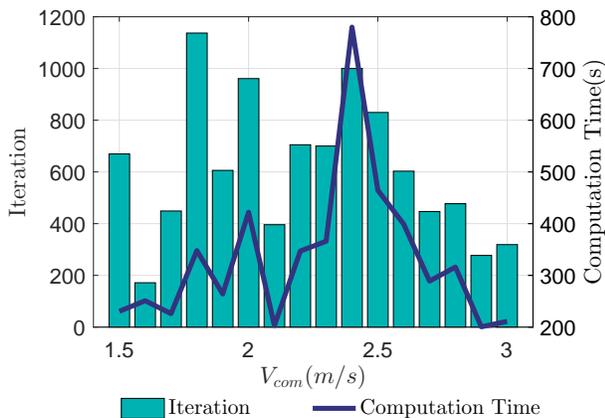


Fig. 7: The computation performance vs running velocity

generated running gaits that satisfy all the physical limitations and constraints, including the HZD condition, fast and reliably. Furthermore, in real world experiments, because of the un-modeled dynamics and system uncertainty, the actual running behavior could show inconsistency with simulation. A major benefit of the computational efficiency is that it offers the flexibility to refine the running behavior and adjust the model practically, paving the way to achieve experimental running.

The Specific Cost of Transport (SCOT), which quantifies the energy efficiency of transporting an object from one location to another, is embedded as the objective by the optimizer. Therefore, another advantage of this framework is the capability of generating robotic locomotion with high energy efficiency. Notably, as shown in Fig. 6a, for a faster running gait, the optimization generated gaits that have higher SCOT, which aligns with the sense that for a particular running pattern, it takes more energy to transport the robot faster.

Eventually, to take advantage of this efficient running gait generation method, we purposely tune the forward velocity constraints to study gaits that runs at different speeds. As shown in Fig. 6c and Fig. 6b, when the pure goal is to

minimize the energy consumption, the optimizer increases both the running frequency and step length to achieve a faster running speed, which agrees with Raibert’s hypothesis and indicates that both are key factors to fast running.

One limitation of this framework is that because of the highly underactuated dynamics, including hybrid system stability constraints in the optimization, i.e., validating the eigenvalues of the linearization of the Poincaré map for such a nonlinear system is not practical (see [11]). Thus, the stability is not enforced or guaranteed by the optimizer. However, as we have checked the magnitude of the eigenvalues of all the gaits afterwards, 24 out of 25 gaits generated from the optimization are stable. Our hypothesis is that because of particular mechanical design, that the underactuated spring is rigidly perpendicular to the foot which stays flat on the ground during stance domain, all the reaction forces exerted on the foot result in upright through the spring. And then the dynamics along x direction resemble single domain fully actuated walking to some extent.

VI. CONCLUSION

In this paper, we documented and tested a framework, built upon a foundation of hybrid zero dynamics (HZD), to generate 3D bipedal running gaits with multiple degrees of underactuation. We used a direct collocation based optimization method to generate HZD motion primitives (i.e. gaits) to reliably handle the added complexity of the underactuated optimization problem. With an eye toward hardware implementation, we incorporated all of the physical limitations of the physical robot DURUS.

We further generated and analyzed multiple running behaviors with different forward speeds from 1.5 m/s to 3.0 m/s. While the optimization and simulation are close to reality in and of itself, the ability to generate gaits quickly simply by varying few constraints can serve as a proxy for more experimental locomotion challenges, such as human-like multi-contact running with foot roll behavior.

REFERENCES

- [1] M. Ahmadi and M. Buehler. Controlled Passive Dynamic Running Experiments With the ARL-Monopod II. *IEEE Trans. on Robotics*, 22(5):974–986, 2006.



Fig. 8: Tiled still images the DURUS running at 2.0 m/s.

- [2] A. D. Ames. Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control*, 59(5):1115–1130, May 2014.
- [3] J. T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*, volume 19. Siam, 2010.
- [4] R. Blickhan. The spring–mass model for running and hopping. *Journal of Biomechanics*, 22(11):1217–1227, 1989.
- [5] S. Collins, A. Ruina, R. Tedrake, and M. Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.
- [6] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body Motion Planning with Simple Dynamics and Full Kinematics. In *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.
- [7] T. Geijtenbeek, M. van de Panne, and a. F. van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6):1–11, nov 2013.
- [8] J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically Stable Walking for Biped Robots: Analysis via Systems with Impulse Effects. *IEEE Trans. on Automatic Control*, 46(1):51–64, Jan. 2001.
- [9] 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.
- [10] J. Guckenheimer and S. Johnson. Planar hybrid systems. In *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*, pages 202–225. Springer Berlin Heidelberg, 1995.
- [11] K. A. Hamed and J. W. Grizzle. Robust event-based stabilization of periodic orbits for hybrid systems: Application to an underactuated 3D bipedal robot. In *American Control Conference (ACC), 2013*, pages 6206–6212, June 2013.
- [12] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of Guidance, Control, and Dynamics*, 10(4):338–342, 1987.
- [13] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. 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 (ICRA)*, 2016.
- [14] Y. Hurmuzlu and D. B. Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, 1994.
- [15] A. E. Martin, D. C. Post, and J. P. Schmiedeler. Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet. *The International Journal of Robotics Research*, 33(7):988–1005, 2014.
- [16] W. C. Martin, A. Wu, and H. Geyer. Robust spring mass model running for a physical bipedal robot. *International Conference on Robotics and Automation*, pages 6307–6312, 2015.
- [17] K. Mombaur. Using optimization to create self-stable human-like running. *Robotica*, 27(03):321–330, 2009.
- [18] B. Morris and J. W. Grizzle. A restricted Poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 4199–4206, Dec 2005.
- [19] H. Park, K. Sreenath, A. Ramezani, and J. W. Grizzle. Switching control design for accommodating large step-down disturbances in bipedal robot walking. In *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, pages 45–50. Ieee, May 2012.
- [20] M. J. Powell, A. Hereid, and A. D. Ames. Speed regulation in 3d robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4803–4810. IEEE, 2013.
- [21] M. Raibert and E. R. Tello. Legged robots that balance. *IEEE Expert*, 1(4):89–89, Nov 1986.
- [22] M. H. Raibert, B. H. Brown, M. Chepponis, J. Koechling, J. Hodgins, D. Dustman, K. W. rennan, D. Barrett, C. Thompson, J. Hebert, W. Lee, and L. Borvansky. Dynamically stable legged locomotion (september 1985-septembers1989), pages 49–77.
- [23] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.
- [24] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki, and A. D. Ames. Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [25] S. Rezaeadeh, C. M. Hubicki, M. Jones, A. Peekema, J. Van Why, A. Abate, and J. Hurst. Spring-mass walking with ATRIAS in 3D: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot. In *Proceedings of the ASME 2015 Dynamic Systems and Control Conference ASME/DSCC 2015*, 2015.
- [26] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.
- [27] M. Spong. Underactuated mechanical systems. In B. Siciliano and K. Valavanis, editors, *Control Problems in Robotics and Automation*, volume 230 of *Lecture Notes in Control and Information Sciences*, pages 135–150. Springer Berlin / Heidelberg, 1998.
- [28] K. Sreenath, H. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.
- [29] K. Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle. Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL. *The International Journal of Robotics Research*, 32(3):324–345, Mar. 2013.
- [30] M. Srinivasan and A. Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072):72–75, 2006.
- [31] A. Wächter and L. 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.
- [32] P. M. Wensing and D. Orin. 3D-SLIP steering for high-speed humanoid turns. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4008–4013. IEEE, 2014.
- [33] 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.
- [34] W. Xi, Y. Yesilevskiy, and C. D. Remy. Selecting gaits for economical locomotion of legged robots. *The International Journal of Robotics Research*, pages 0278–3649, 2015.
- [35] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames. From formal methods to algorithmic implementation of human inspired control on bipedal robots. In *Algorithmic Foundations of Robotics X*, pages 511–526. Springer, 2013.
- [36] H. Zhao, A. Hereid, W. I. Ma, and A. D. Ames. Multi-contact bipedal robotic locomotion. *Robotica*, FirstView:1–35, 2 2016.
- [37] H. Zhao, S. Yadukumar, and A. Ames. Bipedal robotic running with partial hybrid zero dynamics and human-inspired optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1821–1827, Oct 2012.