Realizing Underactuated Bipedal Walking with Torque Controllers via the Ideal Model Resolved Motion Method

Eric Cousineau[1] and Aaron D. Ames[2]

***Abstract*— This paper presents experimentally realized bipedal robotic walking using ideal torque controllers via a novel approach termed the *ideal model resolved motion method (IM-RMM)*, where a system's ideal closed-loop dynamics is integrated forward from the actual state of the hardware to provide desired positions and velocity commands to a PD controller. By combining this method with gaits generated using the Human-Inspired Control framework, walking was realized experimentally on the DURUS platform, designed and built by SRI, and achieved with minimal system identification. For comparison, two controllers, one using feedback linearization and one using Control Lyapunov Function based Quadratic Programs (CLF-QP), both realized through IM-RMM, are compared with a benchmark procedure, the Hybrid Zero Dynamics reconstruction, that is shown to provide reliable walking in literature. The results of both simulations and experiments are presented, with the CLF-QP implemented via IM-RMM resulting in the lowest experimental specific energetic cost of transport of $c_{et} = 0.63$ achieved during sustained walking on the 31.5 kg bipedal robot.**

I. Introduction

Robotic walking presents a wide variety of challenges related to nonlinear control, especially in the domain of underactuation. Underactuated bipedal walking has been achieved on a variety of platforms, including: RABBIT [1], ERNIE [2], MABEL [3], ATRIAS [4] (a.k.a. MARLO [5]), AMBER 1 [6], and during the phase immediately preceding heel strike in multi-contact walking on AMBER 2 [7]. The walking controllers realized on many of these robots utilized an offline nonlinear optimization problem enforcing Hybrid Zero Dynamics (HZD) constraints [1] and were either implemented in real-time with the HZD reconstruction (HZD-R) method [6] or PD control on the outputs with a feedforward torque [5]. For AMBER 1 and AMBER 2, walking was achieved by taking the assumption of remaining on the zero dynamics surface, computing the inverse kinematics for the given point on the surface, and then feeding this command to position control. For MABEL, walking has been achieved both using PD control on the outputs and torque control, through both feedback linearization and Control Lyapunov Function based Quadratic Programs (CLF-QPs) as described in [8].

This paper presents human-inspired walking realized on a new robot, DURUS, shown in Fig. 1, with the hardware and control infrastructure designed and built by SRI International[3], using both HZD-R and *ideal model resolved motion method* (IM-RMM). The novel contribution of this paper is the development and application of IM-RMM in order to implement both feedback linearzation and CLF-QP torque controllers to realize dynamically stable underactuated bipedal walking. In particular, IM-RMM realizes a real-time ideal torque controller by integrating forward the ideal closed-loop dynamics starting from the actual state measured on hardware to produce desired positions and velocities to send to hardware. The motivation for the development of this technology is to have a middle ground between simple but brittle position control and complex but flexible torque control. Therefore, IM-RMM yields an ideal torque controller's second-order behavior while avoiding the extensive system identification necessary for highly accurate torque control.

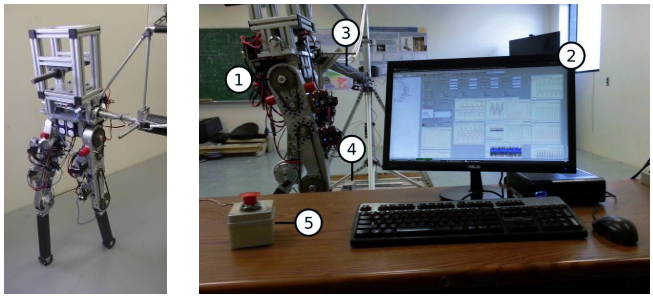


Fig. 1: A snapshot of the DURUS walking in the AMBER Lab. The setup used for operating DURUS, labeled as the following: (1) DURUS with motor power supplied by batteries onboard, (2) the control workstation, (3) the boom, used to constrain DURUS to the saggital plane, (4) logic power supply, and (5) the Emergency-Stop switch.

The nature of resolving a desired position and velocity from a torque control model motivates the name inherited from the resolved motion method [9] used for inverse kinematics. To the authors' knowledge, there is no formal name for a technique such as IM-RMM. In several surveys of robotic control and locomotion, with topics covering operational-space task control [9], feedback linearization [10], ZMP control [9], and local neuromuscular control [11], there are no simple techniques akin to IM-RMM. In the experimental setup of [12], there is a mention of integrating accelerations from a torque controller to yield joint trajectories, but the phrasing and computation times do not make it clear if this is done in simulation then played back on the robot or performed in real-time. There are also methods that

*This work is supported by the SRI International Award W31P4Q-13-C-009 and the Texas Emerging Technology Fund Grant 11062013.

[1]Eric Cousineau is a graduate student in Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA eacousineau@tamu.edu

[2]Aaron Ames is with the Faculty of Mechanical Engineering, Texas A&M University College Station, TX 77843, USA aames@tamu.edu

[3]http://www.sri.com/

incorporate the integration of reduced-order models to produce trajectories for planning. These methods are employed in spring-loaded inverted pendulum (SLIP) models, such as in [13], with an example of code implementation published by Oregon State University's Dynamic Robotics Laboratory in the ATRIAS code repository.[4] Another example is with the simplified linear inverted pendulum model (LIPM), which is used to realize step-based push recovery by planning a trajectory for the center of pressure (CoP) using Model-Predictive Control in [14].

This paper is organized as follows: Sec. II provides a mathematical description of the human-inspired framework and optimization used to generate walking gaits. Sec. III provides a brief overview of the idea controllers considered in this paper: feedback linearization and CLF-QP torque controllers. Sec. IV provides theory for HZD-R and IM-RMM for realizing PD controllers on hardware. Sec. V covers the implementation, with a brief description of the DURUS platform and the deviations from theory used to achieve robust walking. The resulting walking realized on hardware is then discussed in Sec. VI, including the tracking errors and energetic cost of transport. Conclusions and future directions are then stated in Sec. VII.

II. Human-Inspired Control

Human-inspired control was first introduced in [6], building upon the framework of hybrid zero dynamics [1], [15], as a means to define control objectives with canonical representations of human-like walking and, with ideal control, provably realize this walking even on robots undergoing impacts, i.e., hybrid systems. This section gives a brief overview of hybrid system models of walking robots, specific considerations for DURUS, hybrid zero dynamics, and the human-inspired optimization problem that can be used to provide the parameters necessary to realize human-like walking on bipedal robots.

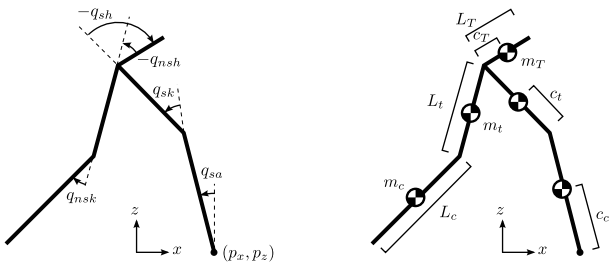
A. Mechanical Hybrid Systems

As shown in [1], [6], [16], symmetric, rigid, underactuated walking can be modeled as a hybrid system with one domain consisting of the continuous dynamics and one discrete transition modeling impacts when the foot strikes the ground. A mechanical system of state space dimension $2n$ has a configuration space $\mathcal{Q} \subset \mathbb{R}^n$, with coordinates $q \in \mathcal{Q}$, and a tangent bundle $T\mathcal{Q} \subset \mathbb{R}^{2n}$, where $x = (q, \dot{q})^T$ represents coordinates describing the state space with $x \in T\mathcal{Q}$. The hybrid system of interest in this paper can be formally defined as the tuple:

$$\mathcal{HC} = (\mathcal{D}, \mathcal{U}, \mathcal{S}, \Delta_R, f, g), \tag{1}$$

where $\mathcal{D} \subset T\mathcal{Q}$ is the domain, $\mathcal{U} \subseteq \mathbb{R}^m$ is set of admissible control values for a robot with m actuators, $u \in \mathcal{U}$ is the vector of inputs, $\mathcal{S} \subset \mathcal{D}$ is the switching surface, the smooth map $\Delta_R : \mathcal{S} \to \mathcal{D}$ is the reset map that occurs at impact, and $\dot{x} = f(x) + g(x)u$ is a control system governing the continuous dynamics of the system [17].

[4] http://bit.ly/1pwHp7S



(a) Coordinates (b) Parameters

Fig. 2: A diagram of the 5-link biped.

TABLE I: Model Parameters

Label	L (m)	m (kg)	c_x (m)	c_z (m)	$\mathcal{I}_y$ (kg · m^2)
Calf (c)	0.46	1.2	0	0.266	0.035
Thigh (t)	0.43	8.8	0	0.218	0.18
Torso (T)	0.13	11.5	0.02	0.16	0.48

The continuous dynamics can be modeled using standard second-order rigid body dynamics as in [10], [18], [19] by taking the Lagrangian and computing the Euler-Lagrange equation, $D(q)\ddot{q} + H(q, \dot{q}) = Bu$, where $D(q)$ is the inertia matrix, $H(q, \dot{q})$ is the bias force (the sum of Coriolis terms, $C(q, \dot{q})\dot{q}$, and potential energy gradient, $G(q)$) and $B \in \mathbb{R}^{n \times m}$ is the actuation matrix. The first-order dynamics, $f(x)$ and $g(x)$, are then derived from these dynamics. Discrete transitions at impact are modeled as a perfectly plastic impact as in [10], [6] given a contact constraint. For a model where stance (support leg) and nonstance (swing leg) coordinates are considered, relabeling is performed after the nonstance foot impacts the ground, which signifies swapping the stance and nonstance legs. A discrete transition occurs at foot strike, i.e., when the system reaches the switching surface $\mathcal{S}$, at which point the impact map is applied: $(q^+, \dot{q}^+) = \Delta_R(q^-, \dot{q}^-)$, mapping pre-impact to post-impact states.

B. Model Considerations

The DURUS robot is rigidly attached to a boom which constrains the robot to walk in a circle with a radius assumed to be large enough to use a planar model, similar to [7]. The floating-base configuration variables, q_e, for this system are given as $q_e = (p, q)^T$, with $q = [q_{sa}, q_{sk}, q_{sh}, q_{nsh}, q_{nsk}]^T$, where "s" prefix stands for the stance (supporting) leg, the "ns" prefix stands for nonstance (swing) leg, and the subfixes "a", "k", and "h" stand for ankle, knee, and hip, respectively, and $p = (p_x, p_z)$ represents the position of the stance foot in the world frame. The stance foot is then assumed to be pinned to the ground, eliminating the need for p and reducing the configuration space to q with $n = 5$ degrees of freedom.

The model parameters for DURUS are shown in Table I, where m is the mass, c_x and c_z are the x- and z-positions of the center of mass relative to the joint origin (noting that the y-axis is the axis of rotation) and $\mathcal{I}_y$ is the rotational moment of inertia about the center of mass for the given link. The coordinates and the parameters are shown in Fig. 2. The boom, shown in the setup in Fig. 1, is modeled as in [1] and [20], by adding its effect on the kinetic energy as an addition to the inertia matrix. The four motors are the m inputs to

the system. The motor rotor inertias – reflected through the gearbox – are incorporated as a decoupled addition to the inertia matrix as

$$D_m = \begin{bmatrix} 0 & 0 \\ 0 & N_m^2 \mathcal{I}_m I_{\times m} \end{bmatrix}, \tag{2}$$

where $\mathcal{I}_m = 3 \cdot 10^{-7} \text{kg} \cdot \text{m}^2$ is the inertia, and $N_m = 30.7$ is the gear ratio.

C. Virtual Constraints

For performing control on the modeled dynamics, one must select an objective, the error between actual outputs and desired outputs, with rank equal to the number of inputs available: m. As in [6], virtual constraints define the control objective of driving actual outputs to desired outputs. In particular, actual outputs are chosen, $y_a : \mathcal{Q} \to \mathbb{R}^m$, with the desired outputs as the canonical walking function:

$$y_{cwf}(t, b) = e^{-b_4 t} \left(b_1 \cos(b_2 t) + b_3 \sin(b_2 t) \right) + b_5, \tag{3}$$

where $y_{cwf} : \mathbb{R}^5 \times \mathbb{R} \to \mathbb{R}$, and $b \in \mathbb{R}^5$ is a vector of parameters. This desired output is used for all outputs, and combined into a vector function: $y_{d,t}(t, \alpha) = [y_{cwf}(t, \alpha_{i+1})]_{i \in O}$, where O is an indexing set for the outputs, and $\alpha = (v_d, \{\alpha_{i+1}\}_{i \in O})$ is the parameter collection tuple for the walking gait, where v_d is the gait's desired walking velocity. Note, in the case of DURUS, $m = 4$, $O = \{1, 2, 3, 4\}$, $\alpha = (v_d, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$.

In order to make the system autonomous as in [6], a monotonic function of the system state, $\tau(q)$, is chosen as the phase variable, which is defined as

$$\tau(q) = \frac{\delta p_{hip,x}(q) - \delta p_{hip,x}(q^+)}{v_d}, \tag{4}$$

where $\delta p_{hip,x}(q) = \left. \frac{\partial p_{hip,x}}{\partial q} \right|_{q=0} q$ is the linearized position of the hip and q^+ is the configuration at the beginning of the step. This phase variable is substituted for t in the desired outputs, yielding $y_d(q, \alpha) = y_{d,t}(\tau(q), \alpha)$, and subsequently the state-based output error: $y(q) = y_a(q) - y_d(q, \alpha)$.

D. Human-Inspired Optimization

Given the formal specification of the control objective through virtual constraints, the human-inspired optimization (HIO) [6] can be used to produce a parameter set and initial condition for steady-state walking. When the control objective is met such that $y = 0$ for all time, the system is said to be on the zero dynamics surface [1], or formally

$$\mathbf{Z}_\alpha = \{x : y(x, \alpha) = 0,\ L_f y(x, \alpha) = 0\}. \tag{5}$$

This surface can be rendered invariant utilizing ideal controllers (and exact models of the system). Yet, due to the fact that walking robots undergo impacts, the parameters of the outputs must be chosen so that $\mathbf{Z}_\alpha$ is invariant through impact; the end result is hybrid zero dynamics (HZD) [1]. In particular, hybrid invariance can be enforced via a constrained nonlinear optimization problem:

$$\begin{aligned} \alpha^* = \underset{\alpha}{\text{argmin}} \quad & \text{Cost}_{c_{mt}^{|\cdot|}}(\alpha) && \text{(HIO)} \\ \text{s.t.} \quad & \Delta_R(\mathcal{S} \cap \mathbf{Z}_\alpha) \subset \mathbf{Z}_\alpha && \text{(HZD)} \\ & \text{Phyiscal Constraints}, \end{aligned}$$

where $\text{Cost}_{c_{mt}^{|\cdot|}}(\alpha)$ computes the 1-norm mechanical cost of transport (see discussion in Sec. VI-A) for one step given the walking gait defined by α. The constraint (HZD) enforces that the parameters yield a zero dynamics surface invariant through impact and the physical constraints bound positions, velocities, and torques of the system to ensure a physically realizable gait. The initial guess of the parameters, α, are the values fit to the mean human data presented in [6].

III. Ideal Torque Control

This section covers the control methods used to realize underactuated bipedal walking. Partial feedback input-output (IO) linearization is presented as a means to achieve exponential convergence in output tracking. Next, Control Lyapunov Function based Quadratic Programs (CLF-QPs) are presented as a method to achieve exponential convergence while minimizing the control objective error subject to a set of physical (torque based) constraints.

A. Feedback Linearization

As presented in [21], partial feedback linearization can be performed using the state-based output errors: $y(q)$. The outputs are of vector relative degree two (the inputs appear in the output acceleration), thus we have $\dot{y}(x) = L_f y(x) = \frac{\partial y}{\partial q} \dot{q}$, and $\ddot{y}(x) = L_f^2 y(x) + L_g L_f y(x) u(x)$, with the Lie derivative operator defined as $L_b a(x) = \frac{\partial a}{\partial x} b(x)$. The output dynamics can be made to evolve according to $\ddot{y} = \mu$, where μ is the output-space input, by utilizing the affine transformation

$$\mu(x) = L_f(x) + \mathcal{A}(x) u(x), \tag{6}$$

where $L_f(x) = L_f^2 y(x)$ and $\mathcal{A}(x) = L_g L_f y(x)$. The input u can then be solved for as

$$u(x) = \mathcal{A}^{-1}(-L_f(x) + \mu(x)). \tag{7}$$

In this case of feedback (IO) linearization, μ is chosen to produce second-order exponential convergence with the response of a critically-damped spring-mass-damper system,

$$\mu(x) = -2\varepsilon \dot{y}(x) - \varepsilon^2 y(x), \tag{8}$$

where $\varepsilon > 0$ is a single control gain for the dynamics of the system. This feedback produces exponential convergence to the control objective as a result of (6) - (8). Note that this controller only stabilizes the output dynamics; as a result, the parameters chosen by the optimization problem (HIO) play an important role in shaping the zero dynamics that govern the overall stability of the system [1], [6].

B. Control Lyapunov Function

Control Lyapunov Function based Quadratic Programs are a formal mechanism to provide control inputs that drive a system toward the control objective, $y \to 0$, in a point-wise optimal fashion while respecting physical constraints of the system (see [20]). Define the output coordinates $\eta(x) = (y(x), \dot{y}(x))^T$ with the dynamics

$$\dot{\eta} = \underbrace{\begin{bmatrix} 0 & I_{p\times p} \\ 0 & 0 \end{bmatrix}}_{F} \eta + \underbrace{\begin{bmatrix} 0 \\ I_{p\times p} \end{bmatrix}}_{G} \mu. \tag{9}$$

Next, a Rapidly Exponentially Stabilizing CLF (RES-CLF) [8] can be defined as follows:

$$V(\eta) = \frac{1}{2}\eta^T P_\varepsilon \eta, \tag{10}$$

with $P_\varepsilon = M_\varepsilon^T P M_\varepsilon$, where

$$M_\varepsilon = \begin{bmatrix} \varepsilon I_{m\times m} & 0 \\ 0 & I_{m\times m} \end{bmatrix} \tag{11}$$

$$P = \frac{1}{\sqrt{3}} I_{2m\times 2m} + F + F^T, \tag{12}$$

where $P \succ 0$ satisfies the continuous time algebraic Riccati equation,

$$F^T P + PF - PGG^T P = -Q, \tag{13}$$

and $Q \succ 0$ selected as $I_{m\times m}$. Then $\dot{V}$ can be computed as an affine transformation of the input, μ, via

$$\dot{V}(\eta) = L_F V(\eta) + L_G V(\eta)\mu \tag{14}$$

$$L_F V(\eta) = \eta^T (F^T P_\varepsilon + P_\varepsilon F)\eta \tag{15}$$

$$L_G V(\eta) = 2\eta^T P_\varepsilon G. \tag{16}$$

Rapid exponential convergence can then be enforced by upper bounding $\dot{V} \leq -\varepsilon\gamma V$, where $\gamma = \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} = \frac{1}{1+\sqrt{3}}$.

The end result of these constructions is an affine inequality in μ that can, therefore, be realized as a constraint in a quadratic program. The affine constraint for the CLF can be reformulated as $A_{CLF}(\eta)\mu \leq b_{CLF}(\eta)$, where

$$A_{CLF}(\eta) = L_G V(\eta) \tag{17}$$

$$b_{CLF}(\eta) = -\varepsilon\gamma V(\eta) - L_F V(\eta). \tag{18}$$

As shown in [17], relaxations can be added in order to allow $V(\eta)$ to grow in order to increase the feasibility in the presence of other constraints such as torque limits. That is, relaxations added via an additional optimization variable, δ, which in this case is a scalar for one CLF. A penalty, W, is placed on the square of this value for the objective function. The relaxation, with scaling N, is added to the inequality yielding $\dot{V} \leq -\varepsilon\gamma V + N\delta$.

The end result of the previous constructions is a quadratic program (QP) aimed at minimizing the output-space input, $\mu(x, u)$, where u is now an argument. The final result is a QP utilizing (6):

$$\begin{aligned} (u^*, \delta^*) = \underset{(u,\delta)}{\operatorname{argmin}} \;& u^T \mathcal{A}^T \mathcal{A} u + 2L_f^T \mathcal{A} u + \delta^T W \delta \\ \text{s.t.} \;& A_{CLF}\mathcal{A}u - N\delta \leq b_{CLF} - A_{CLF}L_f \\ & u_{\min} \leq u \leq u_{\max}. \end{aligned} \tag{19}$$

That is, this QP achieves the desired control objective via the CLF constraint subject to conditions on the torque available on the robot in which to achieve this objective.

IV. Hardware Controllers

This section provides a brief theoretical overview of the Hybrid Zero Dynamics reconstruction (HZD-R) and the ideal model resolved motion method (IM-RMM), the controllers that will be implemented on hardware.

A. HZD-R

For performing initial tests on hardware, it is more conducive to test the behavior by commanding position and velocity for local, decoupled PD controllers for each motor. With HZD-R, desired positions and velocities can be determined using the phase from the actual state of the system and the walking gait generated from (HIO) [4], [6].

If the system is evolving on the HZD surface, it implies that $y_a(x) = y_d(x)$ and $\dot{y}_a(x) = \dot{y}_d(x)$. Inverse kinematics can be performed on the linear outputs to solve for the configuration resulting in

$$q_d(q) = \Phi^{-1} \begin{bmatrix} \delta p_{hip,x}(\tau(q), v_d) \\ y_{d,t}(\tau(q), \alpha) \end{bmatrix} \tag{20}$$

$$\dot{q}_d(q, \dot{q}) = \Phi^{-1} \begin{bmatrix} \delta \dot{p}_{hip,x}(\tau(q), \dot{\tau}(q, \dot{q}), v_d) \\ \dot{y}_{d,t}(\tau(q), \alpha)\dot{\tau}(q, \dot{q}) \end{bmatrix}, \tag{21}$$

where $\Phi = [\frac{\partial p_{hip,x}}{\partial q}; \frac{\partial y_a}{\partial q}]$, $\delta p_{hip,x}(\tau, v_d) = \delta p_{hip,x}(q^+) + v_d\tau$, and $\delta \dot{p}_{hip,x}(\tau, \dot{\tau}, v_d) = v_d\dot{\tau}$ (where τ is given in (4)) These desired positions and velocities, $(q_d, \dot{q}_d)^T$, can then be used as an objective to a PD controller for a physical input, u, as

$$u(q, \dot{q}) = -K_P(q^m - q_d^m(q)) - K_D(\dot{q}^m - \dot{q}_d^m(q)), \tag{22}$$

where $K_P, K_D \in \mathbb{R}^{m\times m}$ are diagonal matrices of the proportional and derivative gains, and q^m corresponds to actuated degrees of freedom. This controller therefore tracks the desired outputs of the walking controller.

B. Ideal Model Resolved Motion Method

Although the HZD-R methodology has been utilized to achieve underactuated robotic walking on a variety of robotic platforms [7], [16], it suffers from some important drawbacks. At a fundamental level, it utilizes the outputs to generate the desired positions of the robot, but as a result it cannot be used to modify the behavior of the robot away from this desired behavior. That is, it can not use information related to the model (beyond its use in the HIO) to modulate the behavior of the robot based upon advanced ideal torque controllers like the CLF-QP (19). We address the main shortcomings of HZD-R through the formulation of an intuitive method, IM-RMM, that allows for the implementation of ideal controllers experimentally without the use of exhaustive system identification.

The core idea behind IM-RMM is to produce desired positions and velocities from an ideal torque controller by integrating forward the ideal dynamics of the system from the actual state of the hardware. More concretely, the closed

loop dynamics can be formed by incorporating the input $u(x)$ into the vector field yielding the autonomous system $\dot{x} = f_{cl}(x) = f(x) + g(x)u(x)$. Integration can then be performed using any valid method for explicitly solving an ordinary differential equation (ODE) given a fixed time step Δt, yielding a trajectory for real-time control.

To provide a brief discrete perspective, given the current cycle k at time $t[k]$, the current state may be defined as $x_a[k] = (q_a[k], \dot{q}_a[k])^T$, and the desired state that will be sent corresponding to this frame is defined as $x_d[k] = (q_d[k],\ \dot{q}_d[k])^T$. With this notation, the flow of data can be defined as

$$x_d[k] = \text{ODE}(f_{cl}(\cdot), x_a[k], t[k], \Delta t), \tag{23}$$

which takes the state, $x_a[k]$ at time $t[k]$ and integrates it forward a time step of Δt, yielding the desired state $x_d[k]$ which is intended to be achieved at $t[k+1] = t[k] + \Delta t$, implying the intent that $x_a[k+1] \to x_d[k]$.

V. Implementation

A. Simulation

The expressions for kinematics and dynamics of the model were defined in Mathematica and then ported to MATLAB code, which were used for the optimization and simulations using the `ode45` integrator, with data sets to test the real-time control code to be implemented. The simulations results for IO via torque control are shown in Fig. 5, Fig. 6, and Fig. 7.

B. Hardware

The robot DURUS, shown in Fig. 1, has four actuators, geared with novel low-friction transmissions from SRI International, placed at both of the knees and both of the hips. At each of the outputs shafts and at the boom, incremental encoders are used to measure the joint rotation, and a strain gauge measured the torque at each of the four joints. There are a total of five microcontrollers: one per joint, controlling the motors and processing sensor data from incremental encoders, absolute encoders, and load cell sensor data, and one in order to process encoder information detecting the rotation of the boom. The microcontrollers and sensors are powered by an off-board logic power supply, while the motors are powered by four on-board batteries. The microcontrollers communicate with a real-time enabled Linux host computer running a real-time process. The real-time process, designed by SRI International, is implemented using the EtherLAB software coupled with MATLAB Simulink Coder, and is set to run at 1 kHz.

The high-level controllers were implemented using C++. The Eigen[5] library was used with minor modifications of Ben Stephen's EQuadProg++[6], selected in lieu of CVXGEN[7] for speed improvements (1 - 12 times faster than CVXGEN) and dynamic resizing. ODEINT[8] was used for the fourth-order Runge-Kutta method (`runge_kutta4`) which could run both IO and the CLF-QP within the 1 kHz duty cycle. For a more complex controller, the simpler Euler method (`euler`) could be employed. The resulting high-level diagram of the position controller options, HZD-R and IO or CLF-QP via IM-RMM, with the gait α generated via the HIO, are shown in Fig. 3.

[5] `http://eigen.tuxfamily.org/`
[6] `http://www.cs.cmu.edu/~bstephe1/`
[7] `http://cvxgen.com/docs/index.html`
[8] `http://www.odeint.com/`

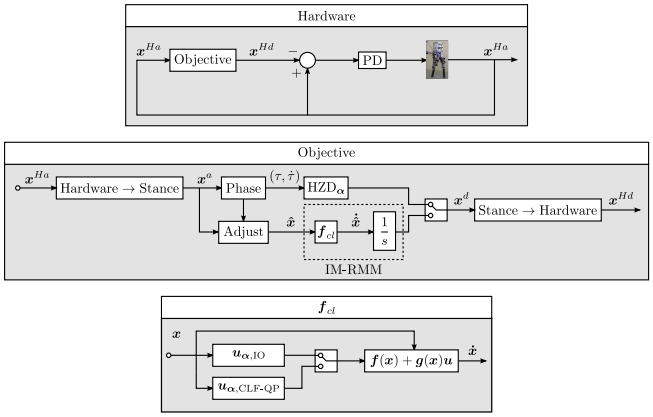


Fig. 3: Diagram of the continuous-time control.

C. Deviations from Theory

With the three control variants, two different loop rates were used for the PD control on the hardware. For HZD-R, a 1 kHz high-level PD controller produced a more compliant command, minimizing feedback oscillations. For IM-RMM, the 10 kHz PD controller at the embedded level was used to achieve tighter tracking at the local position and velocity level. An important component of hardware implementation is the calculation of τ as it appears in (20)-(21). In particular, τ had to be saturated and rate-limited as in [22], with a monotonic constraint. It was also found that using the velocity data read from the system produced desired velocities that caused the system to oscillate. As a temporary fix, $\dot{\tau}$ was computed as a function of τ itself. Due to the nature of feedback linearization, if τ and $\dot{\tau}$ were inconsistent with the actual outputs of the system, it would induce additional oscillations and increase the cost of transport. To resolve this issue, the states were adjusted to be consistent to solve for $(q_{sa}, \dot{q}_{sa})^T$ in terms of $(\tau, \dot{\tau}, q^p, \dot{q}^p)^T$.

VI. Experimental Results

Each controller was run at 1 kHz, with IM-RMM realized using fourth-order Runge-Kutta method, where IO had an average run time of 60μs and the CLF-QP had an average run time of 0.13ms. Data were logged at 200 Hz using EtherLAB's TestManager interface and processed in MATLAB. A video of these experiments are available in [23].

The walking itself shown in Figure 4. A comparison of the tracking for each trial may be seen in Fig. 5 with the limit cycles in Fig. 6. The worst joint-tracking error was found at the nonstance joints with an RMS value of 0.08rad, and the best tracking was found at the stance knee with an RMS value of 0.04rad. This was due to the large

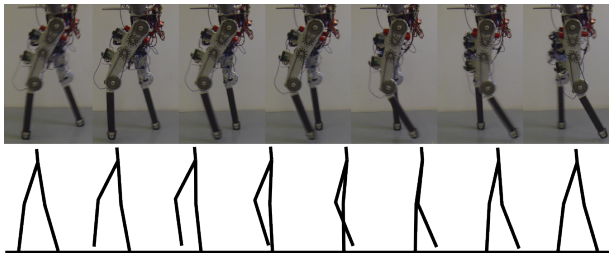
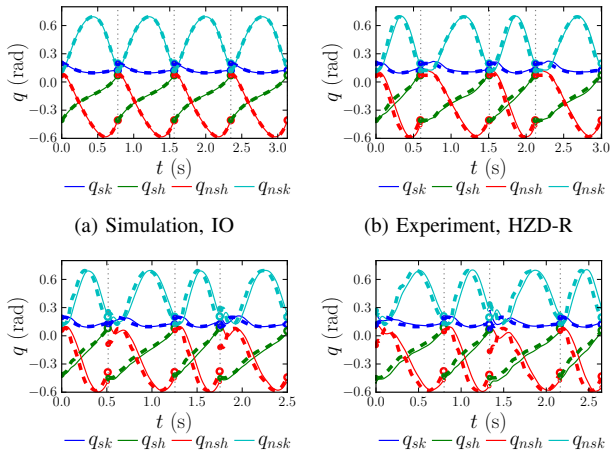


Fig. 4: Walking tiles of the behavior on hardware and in simulation.



(a) Simulation, IO (b) Experiment, HZD-R

(c) Experiment, IM-RMM IO (d) Experiment, IM-RMM CLF

Fig. 5: Comparison of tracking where the solid lines are the actual positions and the dotted lines are the desired positions.

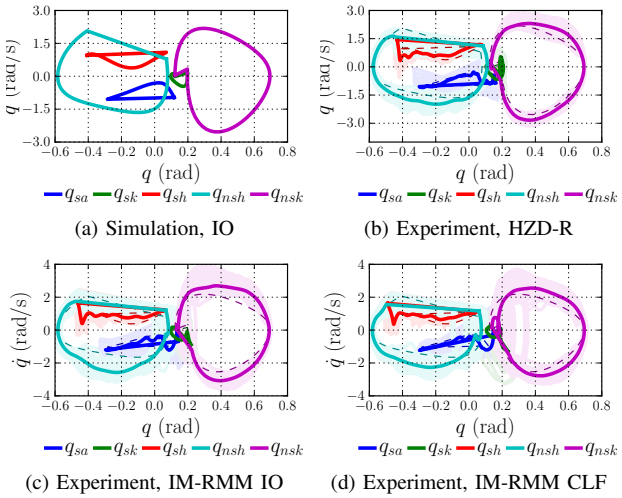
travel for the nonstance joints. For IM-RMM, the joint-tracking errors were computed for the torque controllers using inverse kinematics as a means to provide a common metric for tracking. The tracking error for the resulting PD command was typically less than 0.005rad. The torques are shown in Fig. 7, with a similar magnitude for simulation and experiments.

A. Cost of Transport

The specific cost of electrical transport, c_{et}, is computed as in [24], where the total energy consumed over the product of weight and distance traveled is represented for step i as

$$c_{et,i} = \frac{1}{mgd_i} \int_{t_i^+}^{t_i^-} P_{el} + \sum_{j=1}^{4} I_j(t) V_j(t) dt, \qquad (24)$$

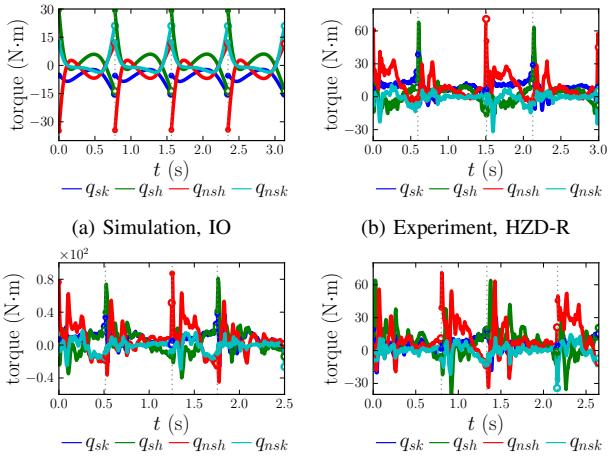
where $P_{el} = 58.25$W is the logic power consumed by the host computer (37.5W, measured using a Kill-a-Watt meter with the real-time process, the GUI, and data logging) and five microcontrollers (20.75W), and $I_j(t)$ and $V_j(t)$ are the currents and voltages recorded for the j^{th} motor. The distance is computed as $d_i = p_{nsf,x}(q_i^-) - p_{nsf,x}(q_i^+)$, where q_i^+ is the post-impact configuration at the beginning of the step, and q_i^- is the pre-impact configuration at the end of the step, and $p_{nsf,x}(q)$ is the x-position of the nonstance foot for the given configuration.



(a) Simulation, IO (b) Experiment, HZD-R

(c) Experiment, IM-RMM IO (d) Experiment, IM-RMM CLF

Fig. 6: Comparison of limit cycles. The dotted limit cycles represent simulation results using feedback linearization. The lighter solid lines represent the raw experimental limit cycles, while the darker solid lines represent the average experimental limit cycles.



(a) Simulation, IO (b) Experiment, HZD-R

(c) Experiment, IM-RMM IO (d) Experiment, IM-RMM CLF

Fig. 7: Comparison of torques measured on the system.

Given that the load cell torques, u_j, and velocities, $\dot{q}_j$ were measured for each actuated joint j, the cost of mechanical transport is computed as

$$c_{mt,i} = \int_{t_i^+}^{t_i^-} \sum_{j=1}^{4} P_j(t) dt, \qquad (25)$$

where $P_j(t)$ is mechanical power required for motor j, using a modification of the three options mentioned in [25]:

1) total power, $c_{mt}^{\pm}$, where $P_j(t) = \text{regen}_{C_m}(u_j \dot{q}_j)$, where $\text{regen}_C(x) = [x]^+ - C[-x]^+$ accounts for regeneration with loss, $[x]^+$ takes the value of x if it is positive or zero otherwise, and $C_m = 1$ is the coefficient of mechanical energy loss;
2) positive-only power, c_{mt}^{+}, where $P_j(t) = [u_j \dot{q}_j]^+$; and
3) positive and negative power, $c_{mt}^{|\cdot|}$, where $P_j(t) = |u_j \dot{q}_j|$, resulting in the 1-norm cost.

TABLE II: Experimental costs of transport and energy consumption, where $\bar{P}_{em}$ is the average amount of electrical power consumed by all of the motors and $r_{em} = W_{em}^{-}/W_{em}^{+}$ is the ratio of negative (regenerated) W_{em}^{-} versus consumed positive (consumed) W_{em}^{+} motor electrical work.

Method	c_{et}	c_{mt}^{+}	$c_{mt}^{\lvert\cdot\rvert}$	$c_{mt}^{\pm}$	$\bar{P}_{em}$(W)	r_{em}
HZD	0.73	0.05	0.14	-0.05	71.77	0.06
IO	0.68	0.08	0.18	-0.03	83.38	0.15
CLF-QP	0.63	0.08	0.18	-0.03	67.72	0.13

The computed costs for each experiment are shown in Table II, with data computed for the inner 80% of the steps are taken. The CLF-QP provides the best cost of transport, followed by feedback linearization (IO), and then the HZD reconstruction. Note that feedback linearization required more motor power on average, but the ratio of regenerated electrical energy for the motors, r_{em}, was significantly higher. Additionally, the torque controllers implemented using IM-RMM yielded faster walking which decreased the amount of logic power consumed while walking. These three controllers were also implemented for a configuration of DURUS without a significant torso mass, thus weighing 10 kg less. These controllers also yielded periodic, robust walking. At one point, walking was tested for approximately 30 to 40 minutes, covering half of a kilometer, and stopping only due to the authors deciding to end the experiment.

VII. Conclusions and Future Challenges

This paper presented a methodology, IM-RMM, that can be used as a method to prototype ideal torque controllers, providing a simple method in which to implement these controllers in a closed-loop fashion that is robust to unmodeled dynamics. In particular, this framework can be used to bypass extensive system identification, and thereby experimentally realize underactuated bipedal walking with only estimated physical parameters. IM-RMM also has the potential to produce commands that yield a cost of transport on the same order of magnitude as existing methods utilize to achieve underactuated walking, e.g., HZD-R. Further work will be aimed at improving the tracking error, implementing true torque controllers to compare performance. Furthermore, the goal is to extend these results, first to compliant systems that may more gracefully handle the impacts, and then to more complex, flat-foot 3D humanoid robots.

ACKNOWLEDGMENT

The authors want to thank SRI International for the design, development, and testing of DURUS, Dr. Benjamin Morris for proposing the initial ideas relating to IM-RMM, and the members of AMBER Lab for their invaluable assistance.

References

[1] 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.

[2] 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*, vol. 33, no. 7, pp. 988–1005, 2014.

[3] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "MABEL, a new robotic bipedal walker and runner," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 2030–2036.

[4] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames, "Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control," *Proceedings of the 17th international conference on Hybrid systems: computation and control - HSCC 2014*, 2014.

[5] B. G. Buss, A. Ramezani, K. A. Hamed, B. A. Griffin, K. S. Galloway, and J. W. Grizzle, "Preliminary walking experiments with underactuated 3D bipedal robot MARLO," *IROS*, 2014.

[6] A. D. Ames, "Human-inspired control of bipedal walking robots," *Automatic Control, IEEE Transactions on*, 2014.

[7] H. Zhao, W. Ma, M. B. Zeagler, and A. D. Ames, "Human-inspired multi-contact locomotion with AMBER2," in *submitted to the International Conference on Cyber-Physical Systems*, 2014.

[8] A. Ames, K. Galloway, K. Sreenath, and J. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *Automatic Control, IEEE Transactions on*, vol. 59, no. 4, pp. 876–891, April 2014.

[9] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.

[10] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, Aug. 2014.

[11] A. Schepelmann, H. Geyer, and M. Taylor, "Development of a testbed for robotic neuromuscular controllers." in *Robotics: Science and Systems*, 2012.

[12] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet, "Dynamic whole-body motion generation under rigid contacts and other unilateral constraints," *Robotics, IEEE Transactions on*, vol. 29, no. 2, pp. 346–362, 2013.

[13] G. Garofalo, C. Ott, and A. Albu-Schaffer, "Walking control of fully actuated robots based on the bipedal slip model," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1456–1463.

[14] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*. IEEE, 2010, pp. 52–59.

[15] K. Sreenath, H.-W. 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*, vol. 30, no. 9, pp. 1170–1193, 2011.

[16] S. N. Yadukumar, M. Pasupuleti, and A. D. Ames, "Human-inspired underactuated bipedal robotic walking with AMBER on flat-ground, up-slope and uneven terrain," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2478–2483.

[17] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs," in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 219–240.

[18] R. Featherstone, *Rigid Body Dynamics Algorithms*, ser. Kluwer international series in engineering and computer science: Robotics. Springer, 2008.

[19] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[20] W.-L. Ma, H.-H. Zhao, S. Kolathaya, and A. D. Ames, "Human-inspired walking via unified pd and impedance control," *ICRA*, 2014.

[21] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer New York, 1999, vol. 10.

[22] E. R. Westervelt, "Toward a coherent framework for the control of planar biped locomotion," Ph.D. dissertation, Citeseer, 2003.

[23] http://youtu.be/6BAzHszOpsA.

[24] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[25] R. Jafari, L. L. Flynn, A. Hellum, and R. Mukherjee, "Energy-conserving gaits for point-foot planar bipeds: A five-dof case study," in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013.