Contact-Implicit Optimization of Locomotion Trajectories for a Quadrupedal Microrobot

Neel Doshi1,2, Kaushik Jayaram1,2, Benjamin Goldberg1,2, Zachary Manchester3, Robert J. Wood1,2, and Scott Kuindersma1

1John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, M.A
Email: ndoshi@g.harvard.edu, scottk@seas.harvard.edu
2Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, M.A.
3Department of Aeronautics and Astronautics, Stanford University, Stanford, C.A.

Abstract—Planning locomotion trajectories for legged microrobots is challenging because of their complex morphology, high frequency passive dynamics, and discontinuous contact interactions with their environment. Consequently, such research is often driven by time-consuming experimental methods. As an alternative, we present a framework for systematically modeling, planning, and controlling legged microrobots. We develop a three-dimensional dynamic model of a 1.5 g quadrupedal microrobot with complexity (e.g., number of degrees of freedom) similar to larger-scale legged robots. We then adapt a recently developed variational contact-implicit trajectory optimization method to generate feasible whole-body locomotion plans for this microrobot, and we demonstrate that these plans can be tracked with simple joint-space controllers. We plan and execute periodic gaits at multiple stride frequencies and on various surfaces. These gaits achieve high per-cycle velocities, including a maximum of 10.87 mm/cycle, which is 15% faster than previously measured velocities for this microrobot. Furthermore, we plan and execute a vertical jump of 9.96 mm, which is 78% of the microrobot’s center-of-mass height. To the best of our knowledge, this is the first end-to-end demonstration of planning and tracking whole-body dynamic locomotion on a millimeter-scale legged microrobot.

I. INTRODUCTION

A. Motivation

Laminate manufacturing processes, such as smart-composite microstructures [40] and printed-circuit MEMS [37] enable rapid and reliable assembly of flexure-based, millimeter-scale devices. One major advantage of these manufacturing techniques is the ability to realize mechanically complex devices with many degrees-of-freedom (DOFs) at the millimeter scale. This has enabled the development of small scale legged microrobots that do not sacrifice dexterity. These legged microrobots leverage favorable inertial scaling [36] to demonstrate remarkable capabilities, including high-speed running [12], jumping [13], and climbing [2]. However, these results have largely been achieved using simplified models [14] and time-consuming experimentation [10] [11] due to the mechanical complexity of the robots and challenges associated with modeling legged systems.

The ability to effectively model and control legged microrobots could augment these experimental approaches and greatly benefit work in this area. We evaluate a model-based optimization method for designing agile locomotion behaviors involving contact that avoids the need for exhaustive experimentation. These model-based tools are increasingly important as microrobots move closer to envisioned applications, including inspection in confined environments [17], search and rescue, and environmental monitoring. This work is also broadly applicable to other systems that interact with the environment through contact, including manipulators and larger legged robots.

B. Related Work

A variety of sophisticated model-based methods have been developed to design trajectories for legged robots, including hybrid [7, 24, 29, 30, 31] and contact-implicit [22, 25, 28, 34] trajectory optimization methods. Contact-implicit methods have the benefit of generating contact sequences as part of the optimization, eliminating the need for a-priori contact mode scheduling. This enables planning for a variety of periodic and aperiodic behaviors. Most methods, however, are limited to first-order integration accuracy, which creates a linear trade-off between the size of the trajectory optimization problem and the resulting trajectory accuracy. For high-dimensional robots with multiple contacts, the generation of accurate motion plans can lead to nonlinear programs with tens of thousands of variables that push the limits of modern solvers. In practice, coarse time discretizations are used to reduce program size, resulting in inaccurate whole-body locomotion plans that are difficult or impossible to realize on a physical robot.

In spite of this drawback, contact-implicit methods are often used to plan for reduced systems, such as zero-moment point or centroidal dynamics models. These low-dimensional trajec-
tories are then stabilized using an inverse dynamics controller that resolve whole-body motions online [32]. Such methods have achieved success on the humanoid Atlas robot [20], and the quadrupedal HyQ robot [38, 39]. Another approach, also implemented on the HyQ robot, uses relaxed or spring-damper contact models specifically calibrated for a given system [21]. Despite some successes, whole-body motion plans produced using general physics-based models have yet to be realized on a physical robot.

Recently, Manchester and Kuindersma [22] developed a variational contact-implicit trajectory optimization scheme that combines ideas from discrete variational mechanics with the complementarity formulation of rigid-body contact to achieve higher-order integration accuracy. Simulations from that work demonstrate that this approach results in more accurate whole-body locomotion plans that can be tracked with simple controllers. In this paper, we extend and apply these methods to plan and track locomotion trajectories on a physical micro-robot.

C. Contributions

Our primary contribution is the development and evaluation of a framework for modeling, planning, and controlling dynamic behaviors for legged microrobots. We develop a full three-dimensional dynamic model of the Harvard Ambulatory MicroRobot (HAMR-VI), a quadrupedal microrobot with eight control inputs, 76 states, and 24 kinematic position constraints [5]. We also adapt a state-of-the-art variational contact-implicit trajectory optimization algorithm to generate physically accurate locomotion plans. Finally, we develop a low-latency estimator and joint-space controller that allow this microrobot to track the generated plans.

Our methods are used to generate nine periodic gaits at three frequencies (2 Hz, 10 Hz, and 30 Hz) on three different surfaces (sandpaper, card-stock, and Teflon). Optimized gaits are shown to move 17% faster per-cycle on average than manually tuned gaits at the same operating conditions. We also execute a gait with an average velocity of 10.87 mm/cycle, the fastest recorded gait for this platform. Finally, we demonstrate the first controlled vertical jump of 9.96 mm, which is 78% of the microrobot’s center-of-mass (COM) height.

D. Paper Organization

The remainder of this paper is organized as follows: we present an overview of the platform and a dynamic model of the system in Sec. II. The trajectory optimization problem for periodic and aperiodic behaviors is formalized in Sec. III. In Sec. IV, we describe the hardware and software used for the locomotion experiments, and we present and discuss the results of these experiments in Sec. V and Sec. VI, respectively. Finally, we draw conclusions and present directions for future research in Sec. VII.

II. DYNAMIC MODEL

A. Platform Overview

HAMR-VI (Fig. 1) is a 4.51 cm long, 1.5 g quadrupedal microrobot with eight independently actuated DOFs. Each leg has two DOFs that are driven by optimal energy density piezoelectric bending actuators (henceforth actuators) [16]. These actuators are controlled with AC voltage signals using a simultaneous drive configuration described by Karpelson et al. [19]. A spherical-five-bar (SFB) transmission (Fig. 2) connects the two actuators to a single leg in a nominally decoupled manner: the swing actuator controls leg-x motion, and the lift actuator controls the leg-z motion. Each SFB transmission has 11 carbon fiber linkages (QA-112, Tohotenax) connected by nine compliant polyimide flexures (Kapton, Dupont) in three parallel kinematic chains.

B. Robot Model

The dynamics of the SFB transmissions are assumed to follow the pseudo-rigid body approximation [15], with the flexures and carbon fiber linkages modeled as pin joints and rigid bodies, respectively. Each flexure is assumed to deflect only in pure bending with its mechanical properties described by a torsional spring and damper that are sized according to the procedure described by Doshi et al. [5]. Given these assumptions, each SFB transmission has two inputs (forces generated by the actuators), and eight generalized coordinates. These include two independent coordinates (actuator tip deflections) and six dependent coordinates (a subset of flexure joint angles) that represent the kinematics of the three parallel chains. Thus, a complete model of the robot has eight inputs, 38 generalized coordinates (76 states), and 24 position constraints.

A three-dimensional computer-aided-design model is developed using SolidWorks (Dassault Systèmes) to capture the kinematics and inertial properties of the microrobot. An open-source SolidWorks to Universal-Robot-Description-Format (URDF) exporter is used to generate an initial URDF model of the microrobot. Actuator forces, joint limits, kinematic-loop constraints, and the mechanics (stiffness and damping) of the flexural joints are then manually incorporated. Furthermore, units are rescaled from SI (seconds, meters, and kilograms) to milliseconds, millimeters, and grams for improved numerical conditioning. The dynamics and control
toolbox Drake \cite{35} computes the terms in the Euler-Lagrange equation from this URDF description using the composite rigid body algorithm \cite{8}.

\[
\frac{d}{dt} D_2 \mathcal{L}(q, \dot{q}) - D_1 \mathcal{L}(q, \dot{q}) + C(q)^T \dot{\lambda} = F^b(q, \dot{q}) + F^{\text{act}}(q)
\]

where \( c(q) = 0 \). (2)

Here \( \mathcal{L} \) is the microrobot’s Lagrangian (including flexural spring energy), \( q \in \mathbb{R}^{38} \) is the vector of generalized coordinates, \( C(q)^T = (\partial c/\partial q)^T \) is the Jacobian mapping constraint forces, \( \lambda \) into generalized coordinates, and the dot superscripts represent time derivatives. \( F^b \) is the vector of generalized flexural damping forces, and \( F^{\text{act}} \) is the vector of generalized actuator forces. Equation (2) enforces the kinematic-loop constraints \( c(q) \). Finally, the slot derivative \( D_i \) indicates partial differentiation with respect to a function’s \( i \)-th argument.

Each actuator is modeled as a force source in parallel with a spring (Fig. 2 inset) to determine \( F^{\text{act}} \). The contribution of the effective mass and damping of the actuator is negligible. An affine approximation of the force source model, and a constant approximation of the spring model developed by Jafferis et al. \cite{16} are used for simplicity. These approximations have been experimentally verified for the range of expected operating voltages (~100-200 V) \cite{5}. The generalized actuator force can then be written as,

\[
F^{\text{act}}(V, q) = B(q)^T (F_s(V, q) - Kq),
\]

where \( F_s \) is the piezoelectric force, \( K \) is the spring stiffness, \( V \) is the AC drive voltage, and \( B^T \) is the Jacobian mapping actuator forces into generalized coordinates. Substituting (3) into (1) and (2) gives a complete set of differential-algebraic equations that capture the dynamics of the microrobot:

\[
\frac{d}{dt} D_2 \mathcal{L}(q, \dot{q}) - D_1 \mathcal{L}(q, \dot{q}) + C(q)^T \dot{\lambda} = F^b(q, \dot{q}) + B(q)^T (F_s(V, q) - Kq)
\]

\[
\dot{c}(q) = 0.
\]

C. Surface Characterization

In addition to modeling the robot’s dynamics, the coefficients of static friction (\( \mu \)) between microrobot’s feet and three surfaces, PTFE (Teflon), card-stock, and 1200 grit sandpaper, are measured. Experiments are conducted using a single leg (Fig. 3) to closely replicate conditions during locomotion. Each surface is placed on an acrylic mounting plate and fastened to a six-axis force sensor (ATI Nano17 Titanium). The single leg is mounted on two micro-positioning stages and centered above the force sensor. Eight trials are run on each surface, and force data is recorded at 100 Hz using MATLAB’s xPC environment (MathWorks, MATLAB R2015a). Force traces for a representative trial are shown in Fig. 3.

The robot’s four feet are modeled as point contacts. The contact forces are decomposed into directions normal and tangential to the running substrate. In the normal direction, collisions must obey a non-penetration constraint:

\[
\phi(q) \geq 0,
\]

where \( \phi(q) \) is a function that returns the signed distance between the four feet and the running substrate. Tangential forces are modeled using Coulomb friction, and they must satisfy the Maximum Dissipation Principle \cite{26}. This states that the friction force instantaneously maximizes the dissipation of kinetic energy, and is a generalization of the 2D concept of friction opposing the direction of motion. Mathematically, this
can be formulated as the following optimization problem:

$$\begin{align*}
\text{minimize} \quad & \dot{q}^T D(q) T b \\
\text{subject to} \quad & ||b|| \leq \mu \gamma, \\
& \text{where } b \text{ is the friction force, } \gamma \text{ is the normal force, } \mu \text{ is the coefficient of friction, and } D^T \text{ is the Jacobian mapping tangential contact forces into generalized coordinates. The norm constraint on } b \text{ ensures that the friction force lies within the Coulomb friction cone.}
\end{align*}$$

Instead of using the linear complementarity formulation described by Stewart and Trinkle [33] to combine the normal [7] and tangential [5] contact constraints with the microrobot dynamics [4], we adapt the variational framework recently developed by Manchester and Kuindersma [22]. The variational framework requires approximating the Lagrange-D’Alembert principle with a quadrature rule before taking variational derivatives [23]. The order of the resulting discrete equations-of-motion depends on the choice on quadrature rule, and we use the midpoint rule for second-order accuracy.

The normal contact constraints [7] and the microrobot’s kinematic constraints [2] are added to the discrete Lagrange-D’Alembert principle with the appropriate Lagrange multipliers, \(\gamma\) (the normal force) and \(\lambda\) (the constraint force), respectively. Taking variations with respect to the generalized coordinates \(q_k\) then leads to the discrete Euler-Lagrange equations for our system:

$$D_2 \mathcal{L}_d(h, q_{k-1}, q_k) + D_1 \mathcal{L}_d(h, q_k, q_{k+1}) + \frac{1}{2} \mathcal{P}_d \mathcal{E}_d(h, q_{k-1}, q_k) + \frac{1}{2} \mathcal{P}_d \mathcal{E}_d(h, q_k, q_{k+1}) + \frac{1}{2} (C(h, q_{k-1}, q_k) + C(h, q_k, q_{k+1}))^T \lambda_k + N(q_{k+1})^T \gamma_k = 0. \tag{9}$$

Here \(\mathcal{L}_d\) is the discrete Lagrangian, defined as the midpoint approximation of the integral of the continuous Lagrangian over a single time step. \(\mathcal{P}_d \mathcal{E}_d\) is an analogous discretization of the generalized external force [4]. Note that, in general, \(\mathcal{L}_d\) and \(\mathcal{P}_d \mathcal{E}_d\) depend on the choice of quadrature rule, and expressions for both are derived for a general system using the midpoint rule in [22]. Finally, \(N(q) = (\partial \phi / \partial q)^T\) is the Jacobian mapping normal contact forces into generalized coordinates.

In addition to [9], solutions must satisfy the following constraints, known as Karush-Kuhn-Tucker (KKT) conditions [3]:

$$\begin{align*}
c_d(q_{k}, q_{k+1}) &= 0 \tag{10} \\
\gamma_k &\geq 0 \\
\phi(q_{k+1}) &\geq 0 \tag{11} \\
\gamma_k^T \phi(q_{k+1}) &= 0,
\end{align*}$$

where [10] ensures that kinematic constraints are enforced, and \(c_d\) is the midpoint approximation of \(c\). The three conditions in [11], collectively known as a complementarity constraint, prevent interpenetration and ensure that contact forces act only when bodies are in contact to push them apart. Such constraints are commonly denoted using the following shorthand notation:

$$0 \leq \gamma_k \perp \phi(q_{k+1}) \geq 0. \tag{12}$$

The KKT optimality conditions for an approximation of the Maximum Dissipation Principle (MDP) [8] are derived and discretized using the midpoint rule in a similar manner. This results in three additional constraints: one equality constraint and two complementarity constraints,

$$\begin{align*}
g_1(h, q_k, q_{k+1}, \psi_k, \eta_k) &= 0 \\
0 &\leq \psi_k \perp g_2(\gamma_k, \beta_k) \geq 0 \\
0 &\leq \beta_k \perp \eta_k \geq 0,
\end{align*} \tag{13}$$

where \(\psi\) and \(\eta\) are Lagrange multipliers and the exact forms of \(g_1\) and \(g_2\) are derived in [22]. The tangential force and corresponding constraints are added to [9] and [11] to complete the dynamics model,

$$\begin{align*}
r(h, q_{k-1}, q_k, q_{k+1}, \gamma_k) + P(q_{k+1})^T \beta_k &= 0 \\
g_1(h, q_k, q_{k+1}, \psi_k, \eta_k) &= 0 \\
c(q_k, q_{k+1}) &= 0 \\
0 &\leq \gamma_k \perp \phi(q_{k+1}) \geq 0. \\
0 &\leq \psi_k \perp g_2(\gamma_k, \beta_k) \geq 0 \\
0 &\leq \beta_k \perp \eta_k \geq 0.
\end{align*} \tag{14}$$

Here, \(r\) is the LHS of [9], and \(\beta\) and \(P\) are defined in [22] based on \(b\) and \(D\). Given \(q_{k-1}\) and \(q_k\), (14) can be solved to find \(\lambda_k, \beta_k, \psi_k, \eta_k,\) and \(q_{k+1}\).

**III. Trajectory Optimization**

The dynamics expressed in [14] are used as constraints in a direct trajectory optimization scheme. The trajectory optimization problem is posed as a standard nonlinear program (NLP) and solved using the constrained optimization solver SNOPT. To ease the numerical difficulties associated with complementarity constraints, we apply a smoothing scheme similar to that used in [9]. For a complementarity constraint of the form \(a \perp b \geq 0\), this smoothing scheme replaces the equality constraint \(a^T b = 0\) with the inequality constraint \(a^T b - s \leq 0\). Here, \(s\) is a non-negative slack variable that alters the feasible region as shown in Fig. 4. We use a single slack variable, \(s_k\), to smooth all complementarity constraints at each knot point. To encourage convergence of solutions towards satisfaction of the true complementarity constraints, we augment the cost function with a term that penalizes
The complete formulation of the trajectory optimization problem is stated as the following NLP:

\[
\begin{align*}
\text{minimize} & \quad J(h, Q, V) + \alpha \sum_{k=1}^{N-1} s_k \\
\text{subject to} & \quad f(h, q_{k-1}, q_k, q_{k+1}, \lambda_k, \psi_k, \eta_k) = 0 \\
& \quad g(q_{k+1}, \lambda_k, \psi_k, \eta_k, s_k) \geq 0 \\
& \quad v_{\min} \leq v_k \leq v_{\max} \\
& \quad q_{\min} \leq q \leq q_{\max},
\end{align*}
\]

where \( J \) is a cost function, \( \alpha \) is a positive scalar weighting parameter, and \( f \) and \( g \) are the equality and inequality constraints in the relaxed version of (14) found in [22]. Furthermore, \( Q \) is the set of robot configurations at the knot points, \( q_k \), \( V \) is the set of control voltages, \( v_k \), and \( C \) is the set of all constraint-related variables, \( \gamma_k, \beta_k, \lambda_k, \psi_k, \eta_k \), and \( s_k \). The input voltages, \( v_k \), are bounded by \( v_{\min} = 0 \) V and \( v_{\max} = 225 \) V to increase actuator lifetime [16]. The flexure joint angles, a subset of \( q_k \), are bounded between \( q_{\min} = -\pi/4 \) and \( q_{\max} = \pi/4 \), which are conservative estimates of the maximum mechanical bend angles. The penalty on the slack variables in the cost function of (15) is an “exact penalty” that has theoretical convergence guarantees with finite values of \( \alpha \) [1]. In practice, we find good convergence with values of \( \alpha \) on the order of \( 10^2 \).

### A. Gait Optimization

We search for gaits near stride frequencies of 2 Hz, 10 Hz, and 30 Hz on three different surfaces: Teflon, card-stock, and 1200 grit sandpaper. The selected frequencies represent different operational regimes for the microrobot as discussed by Goldberg et al. [11]: quasi-static (2 Hz), near the z-natural frequency (10 Hz), and near the roll natural frequency (30 Hz). In addition, the 2 Hz gaits represent a long time horizon for this microrobot as the body natural frequencies are between 10–30 Hz. Finally, these nine gaits cover a wide range of (ground) contact conditions (\( \mu \in [0.29, 1.02] \)), showcasing the versatility of our approach.

The NLP presented in (15) is modified to search for periodic state and input trajectories by enforcing periodicity constraints on all position and velocity decision variables except the \( x \)-position of the floating base. The algorithm minimizes the following cost function that encourages the robot to achieve its maximum theoretical stride length:

\[
J = (x_N - x_g)^T Q (x_N - x_g)^T + \sum_{i=2}^{N-1} \frac{1}{2} \Delta \dot{u}_i^T \Delta \dot{u}_i + \frac{1}{2} \Delta u_i^T \Delta u_i,
\]

where \( x_g \) is a goal state, \( Q \) is a diagonal matrix with \( Q_{ii} \in [10, 50] \) and the remaining diagonal entries equal to one, \( c_i \in [10, 50] \) scales the velocity difference penalty, and \( \Delta \dot{u}_i = \dot{u}_i - \dot{u}_{i-1} \) and \( \Delta u_i = u_i - u_{i-1} \) are the difference between subsequent generalized velocities and control inputs, respectively. To reduce sensitivity to local minima, the optimization is initialized with a heuristic trot gait that achieves roughly periodic locomotion assuming cardstock friction. The goal state is then defined as \( x_g = [10, x_p]^T \), where the first entry corresponds to translating the body forward slightly less than twice the maximum swing displacement in a gait cycle. The periodic subset of the goal state, \( x_p \in \mathbb{R}^7 \), is set to the periodic subset of the final state on the initial trajectory. The difference penalties are applied to discourage chatter in the state and inputs trajectory.

### B. Aperiodic Behaviors

We also used this variational trajectory optimization method to find state and input trajectories for a vertical jump. The following cost function, which encourages the microrobot to jump to a specific height, is minimized:

\[
J = (x_N - x_g)^T Q (x_N - x_g)^T + \sum_{i=1}^{N-1} \frac{1}{2} u_i^T R u_i.
\]

IV. LOCOMOTION EXPERIMENTS

The microrobot’s performance for trajectories found in Sec. III is evaluated in a controlled 20 cm × 20 cm locomotion arena (Fig 5). A proportional-derivative (PD) controller is implemented to track the desired positions of the microrobot’s four legs in the body-fixed frame, and an estimator is developed to provide low latency estimates of the leg positions (Fig. 5). A. Locomotion Arena

Input signals are generated at 2.5 kHz using a MATLAB xPC environment (MathWorks, MATLAB R2015a), and are supplied to the microrobot through a ten-wire tether. Five motion capture cameras (Vicon T040) track the position and orientation of the robot body and the position of the feet at 500 Hz with a latency of 11 ms. In addition, eight piezoelectric encoders (described below) provide low-latency estimates of actuator tip velocities at 2.5 kHz. Finally, a high speed camera (Phantom v7.3) orthogonal to the sagittal plane records the motion of the robot at 500 Hz.

B. Piezoelectric Encoder Dynamics

Each piezoelectric encoder (Fig. 6) provides an estimate of the corresponding actuator’s tip velocity by computing the “mechanical” current (proportional to tip velocity [4]) produced in that actuator. This current, \( i_m \), can be computed by applying Kirchhoff’s law [18] to the measurement circuit in series with a lumped-parameter electrical model of an actuator:

\[
i_m = \frac{V_{\text{sig}} - V_{\text{act}}}{R_s} - \zeta C \dot{V}_{\text{act}} - \frac{V_{\text{act}}}{R}.
\]
The first term on the RHS is the total current drawn by an actuator, which is computed on the xPC target from measurements of the voltages before, \( V_{\text{sig}} \), and after, \( V_{\text{act}} \), a shunt resistor, \( R_s \). The actuator is modeled as a capacitor, \( C \), resistor, \( R \), and current source, \( i_m \), in parallel. The voltage and frequency dependent values of \( R \) and \( C \), and the value of the blocking factor \( \zeta \) have been computed by Jayaram et al. \[13\].

**C. Controller and Estimator Design**

Since the motion capture measurement latency is a significant percentage of the gait period (\( \sim 30\% \) at 30 Hz), a filter is developed to provide low-latency leg position estimates in the body-fixed frame by using the current measurements from the piezoelectric encoders. These currents are scaled by an empirically determined proportionality constant, \( X \), to estimate actuator velocities. For a particular leg, the SFB transmission kinematics are used to define a transformation, \( H \in \mathbb{R}^{3 \times 2} \), from lift and swing actuator velocities to Cartesian leg velocities in the body-fixed frame. This map is used to estimate Cartesian leg velocity in the body-fixed frame, which is integrated over the duration of the latency to achieve low-latency leg position measurements.

The map \[19\] is indexed by achievable leg-\( x \) and leg-\( z \) positions (instead of lift and swing actuator positions) as they are directly estimated,

\[
H(x, z) = \frac{\partial f}{\partial q_l} - \frac{\partial f}{\partial q_d} \left[ \frac{\partial c}{\partial q_d} \right]^{-1} \frac{\partial c}{\partial q_l}.
\]

Here, \( q_l \) are the independent generalized coordinates (actuator positions), \( q_d \) are the dependent generalized coordinates (flexure joint angles), \( c \) is the vector of kinematic loop constraints, and \( f \) is the kinematic mapping from lift and swing actuator positions to Cartesian leg position. We solve an inverse kinematics problem, posed as an NLP, to find values of \( q_l \) and \( q_d \) that result in desired leg-\( x \) and leg-\( z \) values. Twenty-one samples are used in each direction, and \( H(x, z) \) is defined for each leg as a \( 3 \times 2 \) look-up table stored on the xPC target.

This look-up table is then used in the following estimator:

\[
\dot{q}[k]\text{leg} = q[k - \epsilon]\text{leg} + T_s \sum_{\kappa=k-\epsilon}^{k} \frac{1}{2} (\dot{q}[\kappa]\text{leg} - \dot{q}[\kappa-1]\text{leg}) \tag{20}
\]

\[
\hat{q}[k]\text{leg} = H(x[k-1], z[k-1]) \begin{bmatrix} X_s & 0 \\ 0 & X_l \end{bmatrix} \begin{bmatrix} i_{m,n}[k] \\ i_{m,l}[k] \end{bmatrix}. \tag{21}
\]

In \[20\], \( \dot{q}[k]\text{leg} \) is the estimated leg position, \( q[k - \epsilon]\text{leg} \) is leg position measured by the motion capture system with \( \epsilon \) latency, \( T_s \) is the sample rate of the xPC Target, and the summation is a trapezoidal integration of the estimated leg velocity over the duration of the latency. In \[21\], \( H \) is indexed by the previous estimate of the leg-\( x \) and leg-\( z \) positions, and \( i[k]_{m,n} \) and \( i[k]_{m,l} \) are the mechanical currents for the swing and lift actuators, respectively. Finally, \( X_s \) (\( \sim 140\)) and \( X_l \) (\( \sim 120\)) are experimentally determined proportionality constants for the swing and lift actuators, respectively.

These estimated leg positions are used in a simple controller since the swing and lift DOFs are nominally decoupled at pre-resonant drive frequencies \[6\]. This feedback controller simply alters the feed-forward lift and swing input voltages based on the following control law:

\[
V = \begin{bmatrix} K_f^n & 0 \\ 0 & K_f^l \end{bmatrix} V_0 + \begin{bmatrix} K_p^n & 0 \\ 0 & K_p^l \end{bmatrix} e + \begin{bmatrix} K_d^n & 0 \\ 0 & K_d^l \end{bmatrix} \frac{d e}{d t}. \tag{22}
\]

Here \( V = [V_s, V_l]' \) are the input signals provided to the swing and lift actuators, \( V_0 = [V_s^0, V_l^0]' \) are the feed-forward input voltages determined by the trajectory optimization (Sec. \[III\]), and \( e = [e_s, e_l]' \) is the error between the desired and estimated leg-\( x \) and leg-\( z \) positions. Note that the desired leg positions are pre-computed from the optimized state trajectory using the model kinematics. Finally, \( K_p, K_d, \) and \( K_f \) are the proportional, derivative and feed-forward control gains, respectively, and the superscripts \( s \) and \( l \) represent the swing and lift DOF, respectively.
V. RESULTS

We demonstrate a physical implementation of the generated trajectories for nine periodic gaits and a dynamic jumping behavior. In addition, we show that our simple joint-space controller maintains gait timing and tracks the desired leg trajectories in the body-fixed frame for a range of frequencies on a wide variety of surfaces.

A. Gait Optimization

Each gait is executed for 15 cycles with an initial voltage ramp, and the control gains are manually tuned. The feed-forward vary from 0.9 to 1.1, reflecting the utility of the optimized inputs. The mean per-cycle forward velocity for the closed-loop experimental trials (blue) is compared with the optimized trajectory (orange) and a manually tuned gait with the same frequency and average input power (gray) in Fig. 7a.

The 2 Hz closed-loop experimental trajectories achieve an average velocity of 9.77 mm/cycle, which is within 5% of the goal speed of 10 mm/cycle. These gaits also perform 26% better than the manually tuned gaits, which achieve an average velocity of 7.71 mm/cycle. In addition, the planned body pose closely matches that which is executed by the robot (card-stock trial shown in Fig. 8), demonstrating that the model captures most of the microrobot’s salient dynamic properties. The robot also maintains the desired gait timing (front legs depicted in Fig. 7b), and the closed-loop leg trajectories (front left leg depicted in Fig. 7c) closely match the optimized trajectories in air ($z > 0$). However, the robot is unable to push as forcefully into the ground as planned, most likely because of unmodeled serial compliance in the transmissions.

Note that the matching of gait timing and acceptable leg-tracking is also consistent for all 10 Hz trajectories and all 30 Hz trajectories, except on Teflon.

At 10 Hz, the closed-loop experimental trajectories achieve an average velocity of 8.98 mm/cycle, which is close to the desired velocity and 10% faster than the manually tuned gaits.

Figure 7. (a) Mean per-cycle forward velocity for the optimized (orange), closed-loop experimental (blue), and manually tuned (gray) trajectories. Error bars represent one standard deviation (n=15). (b) Optimized (orange) and mean closed-loop experimental (blue) leg height for the front left (solid) and front right (dashed) leg. (c) Optimized (orange) and mean closed-loop experimental (blue) periodic foot trajectories for the front left leg. The blue shaded region represents one standard deviation (n=15).

The card-stock gait at this frequency achieves the fastest per-cycle velocity recorded for this robot, at 10.87 mm/cycle. However, the other two gaits perform slightly worse than expected. This is most likely due to discrepancies between the planned and executed floating base trajectories (see supplementary video). Finally, the average velocity for the 30 Hz closed-loop gaits is slower at 4.24 mm/cycle. The closed-loop experiments on sandpaper and card-stock are still 20% percent faster than the manually tuned gaits and within 20% of the predicted optimized velocities; however, the robot’s performance is poor on Teflon. This frequency (near the roll resonance) is particularly challenging for locomotion using the laterally asymmetric trot gait, and the optimizer has difficulty finding gaits that move at 10 mm/cycle.

B. Aperiodic Behaviors

We also use this method to execute four vertical jumps on a card-stock surface. The average jump height achieved is 9.65±0.21 mm, which is ∼80% of the goal jump height of 12 mm (Fig. 9) and approximately 75% of the robot’s COM height. Manufacturing imperfections lead to lateral asymmetries that cause the robot to roll during the jump, but it does not drift
much from the initial position, with final $x$, $y$ and $yaw$ values of $-3.25\pm0.57$ mm, of $-0.15\pm1.98$ mm, and of $11.12\pm8.56$ deg, respectively.

VI. DISCUSSION

A. Performance Improvements

Our model-based approach yields improvements over previous experimental results collected in [10] [11]. Specifically, an average velocity of $9.21 \pm 1.31$ mm/cycle achieved across the six gaits at 2 and 10 Hz is comparable to the highest previously measured experimental velocity of $9.5$ mm/cycle achieved using careful hand tuning on card-stock surface [10]. Even the three slower 30 Hz gaits move on average 30% faster than previously recorded trots at similar frequencies on a card-stock surface. Additionally, the robot is able to achieve a new highest velocity of $10.87$ mm/cycle, and demonstrate the first controlled vertical jump of $9.96$ mm (78% of COM height). Importantly, these performance improvements were achieved without exhaustive experimentation: tens of experiments were conducted as opposed to hundreds.

B. Quality of Optimized Trajectories

We evaluated the quality of the periodic trajectories by measuring the normalized average slip, $\bar{s}$ defined as:

$$\bar{s} = \frac{1}{4} \int_{t_0}^{t_f} |v_x(t)| dt \int_{i=1}^{4} |v_x^i(t)| dt.$$

(23)

Here, $v_x^i$ is the $x$-velocity of the $ith$ leg in the world-fixed frame, and $v_x$ is the $x$-velocity of the center of mass in the world-fixed frame as measured by the motion capture system. The time interval of interest is bounded by $t_0$ and $t_f$, and $\xi$ is the set of times for which $v_x^i < 0$. Intuitively, normalized slip is the total distance a single leg travels backwards in the world frame normalized by the forward distance traveled by the body, and we present an average value for all four legs. Higher values of $\bar{s}$ indicate increased backwards motion of the legs, decreased forward propulsion, and consequently, reduced performance. The average value of $\bar{s}$ is $0.10 \pm 0.06$ ($n = 9$) for the optimized trajectories. This small magnitude of slip is expected since we demand high performance from the robot. The closed-loop experimental trajectories slipped slightly more, with an average $\bar{s}$ of $0.24 \pm 0.14$ ($n = 9$), and is one of the factors that could have resulted in decreased performance. In addition, the optimizer also finds an intuitive jumping trajectory where all four legs first build spring potential energy, and then simultaneously push into the ground.

C. Limitations

Due to the nonconvexity of the trajectory optimization problem, it is possible for the optimizer to get stuck in poor local optima depending on the initial values of the decision variables. This is most clear at 30 Hz stride frequencies, where other (laterally symmetric) gaits have achieved higher per cycle velocities [11]. Furthermore, our current MATLAB implementation requires several minutes to compute the plans described in the previous sections, and significant speed improvements could be made with a C++ implementation that exploits the sparsity of the problem. Finally, unmodeled serial compliance alters the transmission kinematics at higher frequencies (e.g. near 30 Hz), decreasing the utility of the optimized trajectories and reducing robot performance.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we develop and evaluate a framework for modeling, planning, and controlling dynamic behaviors for legged microbots. We developed a full three-dimensional dynamic model of a complex microbot, and adapt a state-of-the-art variational contact-implicit trajectory optimization algorithm to generate physically accurate whole-body locomotion plans for a variety of operating conditions. These locomotion plans are executed on the robot and result in improved performance, including the fastest recorded per-cycle velocity for this robot and a demonstration of the first controlled vertical jump.

Future work can improve upon the both the existing variational framework and the realization of optimized trajectories on the robot. For microbots with high frequency passive dynamics, the contact mode trajectory often changes over longer timescales than the state and input trajectories. Explicitly encoding this into the NLP formulation by using higher order time-stepping methods could aid convergence and avoid unnecessary contact transitions without sacrificing richness in the contact force trajectories. This would allow for planning over longer horizons, thereby enabling optimization of a broader class of aperiodic behaviors. Furthermore, implementing whole-body locomotion controllers and/or learning-based methods to compensate for unmodeled effects would improve translation to the physical system, resulting in increasingly dynamic behaviors.

ACKNOWLEDGEMENTS

This work is partially funded by the Wyss Institute for Biologically Inspired Engineering and the National Defense Science and Engineering Graduate Fellowship. In addition, the prototypes were enabled by equipment supported by the ARO DURIP program (award #W911NF-13-1-0311).
REFERENCES


January 2014.


